UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

ESCUELA NACIONAL DE ESTUDIOS PROFESIONALES
CAMPUS ARAGÓN

"IMPLEMENTACIÓN DE UN SERVIDOR DE TERMINALES GRÁFICAS BASADO EN EL GNU LINUX TERMINAL SERVER PROYECT EN EL LABORATORIO DE ELECTRÓNICA"

T E S I S

QUE PARA OBTENER EL TÍTULO DE :
INGENIERÍA EN COMPUTACIÓN
P R E S E N T A N :
H E I D I C R U Z C R U Z
I B A R R A M O L I N A M A R I O

ASESOR:
ING. MARTÍN HERNÁNDEZ HERNÁNDEZ

MÉXICO

2005

Autorizo a la Dirección General de [illegible] la UNAM a difundir en formato [illegible] el contenido de mi trabajo recepcional.

NOMBRE: Cruz Cruz Heidi

FECHA: 7/12/09

FIRMA:

Autorizo a la Dirección General de Bibliotecas de la UNAM a difundir en formato electrónico el contenido de mi trabajo recepcional.

NOMBRE: Ibarra Molina Mario

FECHA: 7/12/09

FIRMA:

AGRADECIMIENTOS

A Dios vayan las gracias por su indescriptible dádiva gratuita.

2 Corintios 9:15

Mis padres
Leobardo y Gloria

La corona de los viejos son los nietos, y la hermosura de los hijos son sus padres.

Proverbios 17:6

Mis hermanos
Pedro
Eloísa
Leo

Un compañero verdadero ama en todo tiempo y es un hermano nacido para cuando hay angustia.

Proverbios 17:17

Mario

Existen compañeros dispuestos a hacerse pedazos, pero existe un amigo más apegado que un hermano.

Proverbios 18:24

Heidi

AGRADECIMIENTOS

A Dios porque me dio la sabiduría para iniciar este camino y siempre me mantuvo con el optimismo de saber que la confianza en si mismo es el primer secreto del éxito.

A mis padres Margarita y Mario, porque siempre me han apoyado en los caminos que he emprendido, porque siempre han creído en mi y me han dado la fuerza necesaria para seguir adelante.

A mis hermanos Raquel y Martín, por el apoyo que me brindaron para continuar mis estudios y poder cumplir una meta más en mi vida.

A Heidi por darme la oportunidad de conocerla y estar a su lado, por depositar su confianza en mi y por ser la inspiración para ser mejor persona cada día.

A mis amigos de la preparatoria, de la ENEP y de la DS; por que siempre me brindaron su amistad incondicional y me motivaron a superarme día con día.

A la UNAM por que me brindó la oportunidad de tener una formación profesional y sobre todo humana.

Mario

ÍNDICE

OBJETIVO GENERAL

Implementar el proyecto GNU LTSP (Linux Terminal Server Proyect), que consiste en una plataforma que permite la utilización de terminales gráficas que hagan uso de las aplicaciones de un servidor; brindando una red estable, fácil de administrar y con un nivel de seguridad aceptable.

PLANTEAMIENTO DEL PROBLEMA

Hoy en día la necesidad del uso de nuevos programas exige cada vez, más y mejores capacidades en los sistemas de cómputo, la información necesita ser procesada en un menor tiempo y almacenarse en algún medio con la seguridad de que estará disponible en cualquier momento, para lograrlo es necesario contar con un equipo de cómputo con las mejores características: un buen procesador, suficiente memoria y un disco duro de gran capacidad.

Los fabricantes de computadoras buscan que los usuarios reemplacen continuamente sus equipos por aquellos que lanzan al mercado, como consecuencia el hardware sufre un abaratamiento, debido a esto los diseñadores de software parten de esta consideración para desarrollarlo haciendo un uso indiscriminado de los recursos que ofrece el hardware en la actualidad.

A pesar de que en los últimos años se han disminuido los costos del hardware, la adquisición de computadoras modernas implica una gran inversión, sobre todo para las instituciones de educación pública que no cuentan con los suficientes recursos para comprarlas y esto sin considerar los precios de las licencias de software.

En el caso particular del laboratorio de electrónica, se cuenta con equipo de cómputo funcional, pero que no satisface completamente las necesidades actuales de los alumnos; éstos no cuentan con un espacio propio para almacenar sus archivos, lo que puede provocar que en un determinado momento el disco duro se sature con información inservible, y al no existir un control de permisos para los archivos de otros usuarios y del propio sistema, en cualquier momento pueden ser borrados accidental o intencionalmente.

Al no estar en red, se dificulta la administración y el mantenimiento, por ejemplo al realizar una actualización al sistema o depurar la información de los usuarios se debe hacer máquina por máquina, lo cual provoca pérdida de tiempo.

PROPUESTA

Por esta razón nos interesa implementar un servidor de terminales gráficas en el laboratorio de electrónica, aplicando el Linux Terminal Server Project con la distribución Fedora Core; lo que nos permitirá aprovechar las computadoras de baja capacidad haciéndolas aparentar ser más potentes y confiables.

JUSTIFICACIÓN

Lo que se desea es promover la reutilización de computadoras relativamente antiguas, debido a que en la institución se encuentran equipos pequeños como 486 y Pentium, el objetivo es convertirlos en terminales gráficas dotándolas de la potencia aparente de las PC's modernas. Este tipo de terminales son perfectas para las escuelas porque son fáciles de instalar y requieren poco mantenimiento.

La aplicación de esta tecnología a los ámbitos educativos, permite crear un sistema centralizado de almacenamiento de archivos de los alumnos; cada uno tiene su nombre de usuario y contraseña que le permite solo a él tener acceso a ellos; esto evita un problema muy generalizado en ambientes escolares que es el borrado de archivos o configuraciones del sistema.

Todas las aplicaciones corren en el servidor, los clientes no necesariamente deben tener software ni discos duros locales. Una razón es reducir los costos de mantenimiento del hardware y software en varias computadoras. Al hacer uso de un arranque por red se logra que los archivos sean mantenidos por un servidor central, con la ventaja de poder ser actualizados exclusivamente sobre esta máquina. Desafortunadamente se tiene que trabajar con un disco de arranque porque no se cuenta con tarjetas de red con ROM de inicio, pero esto no cambia de ninguna manera el objetivo del proyecto.

Tal vez en algún momento, los usuarios necesitarán ciertas aplicaciones que solo pueden ser ejecutadas en un sistema operativo diferente al del servidor, como puede ser el caso de Windows; debido a esto, se tiene la opción de que las terminales estén funcionando bajo Windows en un modo local, así como la posibilidad de arrancar Linux por red. Esta opción da otra gran ventaja; cuando una instalación de Windows se corrompe, el administrador del sistema tiene la posibilidad de volver a cargar una instalación de Windows usando el arranque remoto de Linux por red, todo esto por medio de un script automático; dejando atrás el tener que formatear y copiar la instalación en disco manualmente.

Todo esto será posible gracias a la aplicación del sistema operativo Linux, en particular la distribución Fedora Core, lo que implica el uso de programas ofimáticos (procesador de textos, hoja de cálculo, bases de datos, etc.); navegadores (Netscape y Mozilla), hasta la utilización de software educativo. Se decidió utilizar esta distribución por que es una excelente plataforma para el uso de estaciones de trabajo sin disco; además de que se caracteriza por su facilidad de instalación y de configuración.

Tiene un control total de los archivos y configuraciones del sistema, mediante una política muy estricta de permisos propia de todo sistema tipo Unix. Además de que se trata de un sistema que potencialmente es muy seguro pero necesita estar bien configurado y administrado, porque de otra forma puede ser la causa de numerosos problemas relacionados con la seguridad.

INTRODUCCIÓN

El concepto de red ha existido desde los albores de la computación, sus características más representativas son: compartir recursos, minimizar costos y evitar la redundancia de información. La arquitectura de ésta define a las topologías, que puede ser de anillo, bus o estrella y por sus características se definen en redes Ethernet, Token Ring, ARCnet. Cuando se tiene un conjunto de computadoras en un espacio inferior a los 10Km, se está hablando de una red LAN y cuando esta distancia crece y los dispositivos y medios de comunicación son otros, se está hablando de MAN's, WAN's y GAN's.

Cuando se habla de una red se hace referencia a un conjunto de computadoras y periféricos interconectados por algún medio físico. Debido a que las redes están integradas por muchos componentes que deben trabajar juntos para crear una red funcional, donde se agilice el trabajo y se minimicen los costos, para esto en el segundo capítulo se analizarán dichas características para poder aprovecharlos adecuadamente.

A fin de poder determinar mejor qué tecnología de red es la más adecuada, se consideró que todos los equipos se encuentran en la misma habitación y que no se cuenta con muchos recursos económicos como para instalar una red inalámbrica, así que lo más óptimo es Ethernet 10BaseT.

Pensemos un poco en el ¿por qué interesarse en rescatar máquinas relativamente obsoletas?. Actualmente las redes se hacen más y más populares buscando compartir recursos y que mejor si una 486 puede ser reutilizada con ese objetivo; y no solo eso, sino también que pueda aparentar una buena capacidad de procesamiento que se refleja en su rendimiento al ejecutar ciertos programas.

Se desea dotar al laboratorio de un servidor fácil de administrar que cuente con un nivel aceptable de seguridad y sobre todo con un costo de mantenimiento mínimo que sirva como apoyo a los alumnos.

En el capítulo 3, se dará una descripción de todos los pasos necesarios para la implementación del proyecto; en cuanto al servidor se realizará la instalación de los programas y servicios, así como la correcta configuración de los archivos de sistema necesarios. En cuanto a las terminales se describirá el proceso para crear una imagen de arranque, la cual podrá ser escrita en un disquete o en el disco duro; lo que permitirá que las terminales arranquen por medio de la red.

En el capítulo 4, se describirá el proceso de iniciar las terminales ya sea utilizando un disco de arranque o por medio de una imagen grabada en el disco duro y se mencionarán las posibles soluciones a los problemas más comunes, en la instalación y configuración de los servicios en el servidor.

El LTSP es un proyecto de software libre, por tal motivo se necesitan conocer todas las implicaciones de utilizar un código creado bajo esta filosofía; para ello en el primer capítulo se hablará del proyecto GNU y de la Licencia Pública General (GPL). También es importante mencionar la evolución del sistema operativo Linux, el cual fue creado bajo esta misma ideología (crear un sistema operativo completo, sin restricciones de uso y licencias).

¿Pero por qué trabajar con Linux?. Sencillamente porque es de distribución libre y no se necesita pagar licencias para instalarlo, además de que con frecuencia es menos problemático que algunos sistemas comerciales. Puede que no sea el mejor sistema para sus aplicaciones en particular, pero para alguien que está interesado en utilizar las aplicaciones Unix disponibles para Linux, se trata de un sistema de alto "performance".

Algo interesante es que se pueden tener varios programas corriendo simultáneamente sin conflicto alguno, gran parte de ellos son programas del sistema que están inactivos hasta que tienen algún trabajo específico que realizar.

Este proyecto está basado en la tecnología cliente-servidor; por lo que se dará una breve descripción y clasificación de las terminales tipo Unix/Linux. Linux es un sistema de libre distribución que cuenta con todos los archivos/programas necesarios para su funcionamiento en multitud de servidores conectados a Internet. La tarea de reunir todos los archivos/programas necesarios, así como instalarlos puede ser algo bastante complicado y no apto para muchos, por esto mismo nacieron las llamadas distribuciones de Linux, que se dedican a hacer el trabajo difícil para beneficio y comodidad de los usuarios.

Una distribución no es otra cosa, que una recopilación de programas y archivos, organizados y preparados para su instalación. Estas distribuciones se pueden obtener a través de Internet, la diferencia entre estas es la calidad de lo incluido en ellas; algunos programas de instalación pueden ser más intuitivos que otros o más fáciles de utilizar, el kernel/núcleo es el mismo en todas excepto en la versión. Las distribuciones Linux más utilizadas son: Fedora Core, Debian, SuSe, Caldera, Mandrake, Conectiva, entre otras.

Para este proyecto se utilizará Fedora Core en su versión 1.0, y tomando en cuenta que constantemente se están liberando nuevas versiones, será posible realizar un cambio de versión de manera transparente. Además ésta es una de las más desarrolladas y está muy bien soportada por la compañía Red Hat.

Como parte de la implementación de este proyecto, en el último capítulo se realizarán una serie de pruebas que determinen el desempeño actual de la red y proporcionen información acerca de cómo se puede mejorar en un futuro. Además se presentan otras propuestas que sirvan de mejoras para el presente proyecto pero en ningún momento se pretende perder la idea de la recuperación de hardware.

En la parte de seguridad se darán ciertas medidas de prevención para una buena administración del sistema y así evitar ataques que pongan en riesgo el correcto funcionamiento del mismo. No se proporciona software adicional para la cuestión de seguridad ya que Linux contiene un nivel aceptable, además de que nuestra red no requiere de un alto grado de protección ya que solo contendrá trabajos escolares.

Se espera instar a los alumnos al ámbito de la investigación, esperando que estas propuestas sean la pauta de futuros proyectos con el fin de mejorar la red escolar.

ANTECEDENTES

OBJETIVO ESPECÍFICO

Comprender en qué consiste el software libre, así como conocer el origen y evolución del sistema operativo Linux para tener una visión más amplia de cómo surge y para qué se puede utilizar el proyecto LTSP; además el analizar las características y aplicaciones de los tipos de terminales, será fundamental para sacar el máximo provecho de éstas.

1

INTRODUCCIÓN

Es importante definir claramente el concepto de software libre para dar a conocer los derechos y libertades de un usuario cuando adquiere un software de este tipo; familiarizarse con el sistema operativo Linux permitirá saber con qué herramientas se puede contar para la implementación de este proyecto, también se aclarará el termino de distribución Linux centrándose en Fedora Core debido a que es muy accesible y fácil de actualizar, ya que con otras distribuciones la actualización es más laboriosa ó incluso puede ser imposible si no se cuenta con los conocimientos necesarios; tampoco conviene que cuando se realice una actualización completa se tenga que formatear el disco duro, lo que implica la pérdida total de la información.

Además cuando se utiliza un sistema operativo, lo que se desea es minimizar la cantidad de trabajo que se tiene que llevar a cabo para concluir alguna tarea o proceso, también es importante que sea un sistema estable y que pueda correr en muchas plataformas.

Para esto el proyecto GNU se inició como un esfuerzo para escribir un avanzado sistema operativo portable y robusto con el estilo de Unix; que pudiera correr en una gran variedad de arquitecturas, no sólo en PC's Intel o Macintosh.

Después de conocer la potencialidad que el software libre ofrece se hablará del proyecto LTSP que permite trabajar con terminales configuradas bajo un modo gráfico y no solo en modo texto. Es verdad que una terminal en modo texto consume muy pocos recursos del sistema además de utilizar poco ancho de banda al acceder de forma remota, pero por otro lado tienen una limitada capacidad expresiva, por lo que es preferible una terminal en modo gráfico para brindar un ambiente más amigable y no solo eso, también se puede contar con una mayor robustez que en una interfaz en modo texto.

1.1 Software libre

Richard Mathew Stallman fue quien inició la concepción actual del término "software libre", y de hecho ha sido la persona más importante dentro de este movimiento. A los 18 años, ingresó al laboratorio de inteligencia artificial del Instituto Tecnológico de Massachusetts, en esa época el software se compartía sin ningún problema así que él se formó dentro de una comunidad que compartía todo.

Al comenzar la década de los 80's se produjeron algunos hechos que lo motivaron a tomar la decisión de iniciar un movimiento en contra del software propietario, que a diferencia de éste se fundamentaba en la idea de compartir el código fuente del software.

Richard utilizó la definición de software libre para especificar de forma clara, lo que debe cumplir un programa de software concreto para que se le considere software libre, él determinó que el software libre era un asunto de libertad y no de precio.

Para entender el concepto, se tiene que pensar en "libre" como libertad de expresión; se debe tomar en cuenta que en inglés la palabra free tiene dos significados "libre" y "gratis", lo que comúnmente da lugar a cierta confusión, en cuanto a lo que el término software libre hace referencia.

Software libre se refiere a la libertad de los usuarios para ejecutar, copiar, distribuir, estudiar, cambiar y mejorar el software. De modo más preciso, se refiere a cuatro libertades de los usuarios del software: [1]

- **Libertad 0**. La libertad de usar el programa, con cualquier propósito.
- **Libertad 1**. La libertad de estudiar cómo funciona el programa, y adaptarlo a las necesidades propias. El acceso al código fuente es una condición previa para esto.
- **Libertad 2**. La libertad de distribuir copias, con lo que se puede ayudar a otras personas.
- **Libertad 3**. La libertad de modificar el programa y hacer públicas las mejoras a los demás, de modo que toda la comunidad se beneficie. El acceso al código fuente es un requisito previo para esto.

Un programa es software libre si los usuarios tienen todas estas libertades. Así pues, se tiene la libertad de distribuir copias, sea con o sin modificaciones, sea gratis o cobrando una cantidad por la distribución, a cualquiera y a cualquier lugar. El ser libre de hacer esto significa entre otras cosas que no se tiene que pedir o pagar permisos.

---

[1] Pág. web de la FSF, Sección de Filosofía http://www.fsf.org/philosophy/free-sw.es.html

1.1.1 Proyecto GNU

El proyecto GNU es un intento de escribir un sistema operativo portable, avanzado, gratuito y de código fuente público, y de aspecto similar a Unix. Richard Stallman comenzó a formar una comunidad de programadores que se dedicaban a desarrollar software libre, pero necesitaba de un sistema operativo que también siguiera esta misma ideología. Así que decidió crear un sistema compatible con Unix al cual le llamó GNU. El proyecto GNU es un acrónimo recursivo para GNU is Not Unix (GNU No es Unix).[2]

Este sistema tendría la capacidad de correr programas Unix, pero esto no significaba que tendría que ser idéntico a este; ya que se le harían las mejoras convenientes basadas en la experiencia que tenía con otros sistemas operativos.

Richard siempre tuvo problemas con el núcleo o kernel de su sistema operativo, él indicaba que ya poseía un núcleo pero que faltaba mucho para que pudiera ser liberado. En 1990 el sistema GNU estaba casi completo y el único componente faltante era precisamente el núcleo, el cual nunca llegó a ser completamente funcional y debido a esto el proyecto empezado por Linus Torvalds llamado Linux, fue de gran ayuda para continuar con el proyecto GNU, aportando el kernel de su sistema operativo.

Actualmente son bastantes las variantes del sistema operativo GNU que utilizan el kernel Linux; aunque estos sistemas son frecuentemente referidos como "Linux", deberían llamarse más propiamente sistemas GNU/Linux.

1.1.2 Fundación para el software libre (FSF)

A medida que crecía el interés por el uso del editor de textos GNU EMACS, creado por Richard, otras personas se involucraron en el proyecto GNU. Entonces nació la fundación para el software libre, esta fundación se dedicaba a eliminar las restricciones sobre el copiado, redistribución, entendimiento y modificación de programas de computadoras.

Muchas organizaciones distribuían cualquier software libre que estuviera disponible. En cambio, la FSF se concentraba en desarrollarlo, y en hacer de éste un sistema funcional el cual pudiera eliminar la necesidad de usar el software propietario.

Además de desarrollar GNU, la FSF distribuía copias de software GNU y manuales por un costo de distribución, y aceptaba donaciones deducibles de impuestos, para apoyar el desarrollo de su software.

---

[2] Pág. web del Proyecto GNU, Sección de Historia http://www.gnu.org/gnu/thegnuproject.es.html

1.1.3 Licencias

Con el marco legal actual la licencia bajo la que se distribuye un programa delimita exactamente los derechos que tienen sobre él sus usuarios. Por ejemplo, en la mayoría de los programas propietarios la licencia priva al usuario de los derechos de copia, modificación, préstamo, alquiler, uso en varias máquinas, etc. De hecho, las licencias suelen especificar que la propietaria del programa es la empresa creadora del mismo, la cual simplemente vende derechos restringidos para el uso del programa.

En el mundo del software libre, la licencia bajo la que se distribuye un programa también es de gran importancia. Normalmente, las condiciones de las licencias de software libre son el resultado de un compromiso entre varios objetivos hasta cierto punto contrapuestos. Entre ellos, pueden citarse los siguientes:[3]

- Garantizar algunas libertades básicas (de redistribución, de modificación, de uso) a los usuarios.
- Asegurar algunas condiciones impuestas por los autores (cita de su nombre en trabajos derivados, etc).
- Procurar que los trabajos derivados sean también software libre.

Los autores pueden elegir proteger su software con distintas licencias según el grado con que quieran cumplir cada uno de estos objetivos, y los detalles que quieran asegurar. De hecho, el autor de un programa suele elegir con mucho cuidado la licencia bajo la que lo distribuye. Por otro lado, los usuarios y especialmente quienes redistribuyen o modifiquen el software, deben analizar con detenimiento la licencia del mismo.

Un punto clave dentro del desarrollo del proyecto GNU, fue la creación de la licencia GPL. Fue un gran éxito para Stallman y su gente lograr darle un marco legal, al movimiento que estaban forjando.

1.1.3.1 Licencia Pública General (GPL)

Este es un tipo de licencia sobre la propiedad intelectual en la cual únicamente se exige que aquellos desarrollos hechos con material licenciado bajo GPL sean a su vez GPL, es decir; se trata de proteger la no ocultación de código.

Antes de la GPL, había un vacío legal ya que la FSF no tenía un instrumento jurídico que le permitiera proteger de la manera que ellos deseaban al software de su propiedad. A partir de esta licencia surgió el concepto de "copyleft".

---

[3] Pág. web del Proyecto GNU, Sección de Licencias http://www.gnu.org/copyleft/gpl.html

Copyleft dice que cualquiera que redistribuye el software, con o sin cambios, debe dar la libertad de copiarlo y modificarlo. También provee un incentivo para que otros programadores aporten ideas al software libre.

Las licencias que cubren la mayor parte del software están diseñadas para quitar la libertad de compartirlo y modificarlo. La licencia pública general pretende garantizar lo contrario para asegurar que el software es libre para todos sus usuarios. Esta Licencia se aplica a la mayor parte del software de la free software foundation y a cualquier otro programa si sus autores se comprometen a utilizarla.[4]

1.1.3.2 Otras licencias

Las dos categorías principales de licencias de software libre son con copyleft y sin copyleft; las licencias con copyleft tales como la GPL insisten en que las versiones modificadas de algún programa creado bajo esta licencia deben ser también software libre. Sin embargo, las licencias sin copyleft no protegen al software para que sea totalmente libre, ya que incluyen ciertas cláusulas donde se especifica hasta que punto el software modificado es libre. Se recomienda utilizar licencias con copyleft porque protege la libertad de todos los usuarios, pero el software sin copyleft aún es software libre, y útil para toda la comunidad.

Hay unas cuantas variantes de licencias simples de software libre sin copyleft, entre las cuales se incluyen la licencia de X10, la licencia X11/XFree86, la licencia FreeBSD, y las licencias BSD (Berkeley System Distribution); la mayoría de ellas son equivalentes excepto en detalles de como están expresadas.

1.2 Linux

El sistema operativo Linux se inició como un proyecto universitario de Linus Torvalds, un estudiante de la universidad tecnológica de Helsinki. En un principio surgió como una idea para desarrollar un sistema operativo, que basado en MINIX[5] y siguiendo el estándar POSIX mejorara a éste. Actualmente es el fruto del trabajo de miles de voluntarios de todo el mundo, que han contribuido a mejorar y añadir nuevas características al sistema.

Paralelamente a este proyecto se originó el proyecto GNU, bajo el cual se desarrollan miles de aplicaciones y utilidades. El sistema Linux fue incluido en dicho

---

[4] FLORIANO Blanco Lino, GALÁN Galán Susana, Linux Al día en una hora, Ed. Anaya Multimedia, España, 1999, p.35

[5] Pequeño sistema operativo escrito por Andrew Tannenbaum con el fin de simular el funcionamiento de un sistema Unix en una PC; que solía ser usado por los estudiantes junto con su código fuente.

proyecto y por lo tanto, actualmente se habla del sistema GNU/Linux al referirnos al sistema completo (sistema y aplicaciones que lo acompañan).

La colaboración de un número cada vez mayor de programadores, aficionados y expertos en Unix, fue fundamental para llevar a cabo el rápido desarrollo que ha experimentado; y desde todo el mundo surgen las aportaciones que constantemente seguirán mejorando y ampliando las capacidades de su kernel.

Linus Torvalds terminó lo que llamó versión 1.0 a principios de 1992, hasta entonces había desarrollado varias versiones iniciales a las que fue aportando la funcionalidad básica. En la actualidad Linux se conoce como un clon de Unix debido a que se basó en la filosofía de introducir las funcionalidades en pequeñas partes (módulos), si aparecen nuevas utilidades éstas pueden ser integradas al espacio de trabajo sin tener que cambiar el resto de los programas que suelen usarse. Es por eso que varios millones de personas lo utilizan en todo el mundo, ya que se ha convertido en un movimiento al que cada vez, mayor número de grandes compañías se están uniendo, aportando soluciones tanto comerciales como bajo licencia GPL (Sun Microsystems, IBM, etc.).

Como Linux es un proyecto de desarrollo abierto, todas las nuevas versiones que vayan apareciendo estarán disponibles para el público, sean o no estables. Sin embargo, para ayudar a la gente a reconocer si una versión es o no estable, como ejemplo se ha acordado lo siguiente: las versiones 1.x.y en las que "x" sea par son versiones estables y el incremento de "y" implica la corrección de algún error. Por lo tanto de la versión 1.2.2 a la 1.2.3 sólo hay corrección de errores, pero ninguna característica nueva. Las versiones 1.x.y con "x" impar son betas para los desarrolladores.

1.2.1 Distribuciones

Uno de los primeros conceptos que aparecen al iniciarse en Linux es el de distribución. Una distribución es un agrupamiento del núcleo del sistema operativo Linux y otra serie de aplicaciones de uso general. En principio las empresas que desarrollan las distribuciones de Linux están en su derecho al cobrar una cierta cantidad por el software que ofrecen, aunque en la mayor parte de las ocasiones se pueden conseguir estas distribuciones desde Internet, de revistas o de otras personas, siendo todas estas formas gratuitas y legales.

Las distribuciones más conocidas son Fedora Core (antes Red Hat), Debian, Mandrake, Slackware, SuSE y Corel Linux, todas ellas incluyen el software más reciente y utilizado que incluye compiladores de C/C++, editores de texto, juegos, programas para el acceso a Internet, así como el entorno gráfico de Linux.

1.2.1.1 Fedora Core

Red Hat como organización, está formada por un grupo de programadores con base en Carolina del Norte, desde sus inicios su objetivo ha sido el de hacer fácil la instalación y utilización de Linux. Al igual que otros grupos parecidos, trata de reunir todos los programas necesarios en una distribución completa y simplificar los aspectos más complejos que pueden aparecer durante la instalación y arranque de un nuevo sistema operativo.

La distribución Red Hat Linux estaba enfocada principalmente a usuarios comunes, como de oficina o de escuelas, pero poco a poco fue introduciéndose en ambientes empresariales debido a que se trataba de un sistema operativo cada vez más robusto y confiable. Desafortunadamente las ganancias eran pocas y había que cubrir los sueldos de los programadores y demás personal que trabajaba en el proyecto, por lo que la empresa tuvo que tomar la decisión de comercializar sus productos; y lanzó al mercado su nuevo sistema operativo enfocado a ambientes empresariales: "Red Hat Enterprise Linux".

Sin embargo, la empresa Red Hat comprometida desde el comienzo con la difusión del software libre y su filosofía, creo un nuevo proyecto, el cual es la versión libre de su sistema operativo, al cual llamó Fedora Core. Fedora Core al igual que su antecesor Red Hat, es una distribución Linux muy diferente a otras, en vez de ser una fotocopia de un disco duro o un conjunto de disquetes a partir de los cuales se pueden formar las diferentes partes de un sistema operativo, está basada desde el principio en paquetes, lo que permite que el sistema pueda ser fácilmente actualizado. Con otras distribuciones, la actualización resulta ser en ocasiones muy complicada.

Desde la introducción de Red Hat en 1994 se han logrado cosas importantes, por ejemplo: el soporte para cualquier tipo de plataforma hardware, mayor fiabilidad del sistema y el uso creciente de Linux por parte de muchas empresas en todo el mundo. Pero muchas cosas han permanecido iguales y Linux sigue siendo un sistema operativo desarrollado por muchas personas en todo el mundo y Linus Torvalds continúa formando parte de este proyecto.

Fedora Core funciona principalmente sobre tres plataformas: PC Intel compatibles, Digital Alpha y Sun SPARC. Los códigos unificados y las ventajas aportadas por la tecnología RPM dan la posibilidad de hacer disponible ésta distribución para cualquier tipo de sistema con un esfuerzo mínimo. Esto en la práctica permite a los usuarios gestionar y transferir el software entre estas plataformas de la manera más sencilla posible.

Además Fedora Core es fácil de administrar, ya que viene con un conjunto de herramientas para el administrador del sistema que reduce las tareas de gestión diarias. Aunado a esto, un completo código fuente viene incluido en todos los componentes libres del sistema.

1.3 Terminales

En Linux y otros sistemas operativos similares se considera la posibilidad de que una computadora tenga varios usuarios trabajando simultáneamente. Cada dispositivo que permite a un usuario interactuar con una máquina se le llama terminal; hay terminales de entrada como por ejemplo un lector de código de barras, los ya obsoletos lectores de tarjetas perforadas, terminales de salida como por ejemplo una impresora, y terminales de entrada-salida como las clásicas terminales de pantalla. Hay otras clases de terminales, pero el tipo más utilizado es el formado por pantalla y teclado, conocidas también como terminales tontas. Dentro de esta categoría hay cientos de modelos distintos y en Linux se contempla la posibilidad de trabajar casi con cualquier tipo de terminal.

Las terminales tontas recibieron ese nombre por el hecho de que no se realizaba ningún procesamiento en la terminal misma, sino que se utilizaba para enviar datos a la computadora anfitriona (o host) por medio del teclado y para recibirlos por medio de la pantalla. En la actualidad, desde el punto de vista conceptual, la arquitectura de una terminal tonta se enfoca principalmente al software y consiste en transmitir la interfaz gráfica de una aplicación o programa que se ejecuta en un servidor hacia la pantalla del usuario final.

El concepto de terminal tonta, se aplica principalmente a las computadoras o dispositivos que constan simplemente de un monitor, teclado, ratón y un pequeño CPU con especificaciones mínimas de memoria y de procesador, así como tarjeta de red integrada. Usualmente las terminales tontas pueden configurarse de dos formas:

- Terminales modo texto (ASCII)
- Terminales gráficas

1.3.1 Terminales modo texto

Antes de la llegada de los entornos gráficos se trabajaba fundamentalmente en lo que se conocía como modo texto o consola, éste tipo de interfaz se sigue utilizando en las terminales X, en las consolas de sistemas Unix o en una consola de MSDOS, la interfaz que se le ofrece al usuario espera que éste introduzca comandos a ser procesados y pueda mostrarle resultados.

A muchos usuarios les gusta todavía este tipo de interfaz, que no necesita para su uso dispositivos como el ratón, y que permite centrarse muy bien en las labores que se están llevando a cabo. Además de que consume pocos recursos del sistema y es sencillo acceder de forma remota utilizando muy poco ancho de banda.

Uno de los primeros problemas que se encuentran a la hora de utilizar una interfaz modo texto es su limitada capacidad expresiva. Al principio se disponía de un tipo de texto sin colores y con unos caracteres limitados. Poco a poco se fue enriqueciendo las posibilidades de estas terminales por ejemplo, en las actuales terminales gráficas se muestran colores y diferentes tipos de letra.

1.3.2 Terminales Gráficas

Las terminales gráficas comenzaron a desarrollarse para uso profesional y técnico, en actividades como la ingeniería donde se necesitaban altas prestaciones en cuanto a resolución y colores. La generalización del uso de las terminales gráficas, debido a su estandarización, ha llevado a sustituir muchas terminales, incluso las terminales ASCII, que corresponden a las de prestaciones más bajas.

Por otro lado, la sencillez de uso y robustez que se pueden alcanzar con una interfaz gráfica es mucho mayor que en un modo texto. Por ejemplo, mediante el uso de menús se logra que el usuario pueda ejecutar comandos de forma guiada, sin necesidad de conocer la sintaxis concreta del mismo. Incluso los comandos más utilizados se pueden incluir en una barra de herramientas de fácil acceso para un uso más rápido de la interfaz. Y algunas de las grandes ventajas del modo texto, como son la accesibilidad para personas discapacitadas o el uso por completo del teclado sin necesidad de perder el tiempo apuntando con el ratón, se puede lograr igualmente diseñando de forma correcta la interfaz gráfica.

Las terminales gráficas son computadoras que soportan el sistema X Window[6], éste suministra una interfaz amigable a los usuarios, a base de la utilización de ventanas y el uso del ratón, actuando como intermediario entre la aplicación y el sistema de presentación. La comunicación entre la terminal gráfica y el procesador correspondiente en un sistema X Window queda definida en el protocolo estandarizado X.11; el juego de bibliotecas que utiliza, también estandarizadas son conocidas como XLib.

Los orígenes de X Window se remontan a mediados de las década de los ochenta, durante el diseño del entorno de ventanas W para el sistema operativo V, sobre el que se estaba trabajando en la Universidad de Stanford. Paralelamente el Proyecto Athena, cuyo desarrollo se llevaba a cabo en el Instituto Tecnológico de Massachusetts; tenía como objetivo diseñar e implantar una interfaz gráfica independiente de una plataforma de hardware concreta y que fuera capaz de proporcionar servicios a través de una red de computadoras. Ambos proyectos se fusionaron en uno solo, y así en 1986, surgió la primera versión de X Window.[7]

---

[6] Hace referencia a la interfaz gráfica de usuario, es el entorno gráfico de los sistemas basados en Unix.
[7] CATALINA Gallego M., CATALINA Gallego A., Unix/Linux Inicialización y Referencia, Ed. Mc. Graw Hill, España, 1999, p.99

El sistema de ventanas X Window está basado en el modelo cliente – servidor el cual es muy utilizado en entornos de red. Su principio de funcionamiento es muy sencillo: un servidor es un programa que ofrece un cierto servicio a otros programas llamados clientes, los cuales emplean al servidor para llevar a cabo ciertas funciones. El servidor X es el programa responsable del control de los dispositivos que se utilizarán en la interfaz del usuario, esto es: el monitor, el teclado y el ratón. Además es quien recibe las señales correspondientes a las teclas pulsadas por el usuario o los movimientos del ratón.

Los clientes X son todos aquellos programas que utilizan el sistema X para interactuar con los usuarios, para ello establecen una conexión con el servidor X. A través de ésta indican al servidor que es lo que quieren representar en la pantalla, y reciben de éste las órdenes que el usuario introduce a través del teclado y el ratón. El servidor X funciona de esta manera como un intermediario entre los dispositivos y las aplicaciones.

El gestor de ventanas es una aplicación cliente que suministra al sistema X un aspecto y características consistentes para todas las ventanas que se muestran en la pantalla. Controla la forma de los botones, los títulos y bordes de las ventanas, la apariencia general del escritorio, etc. La modularidad, característica fundamental de los sistemas Unix/Linux, también está presente en X Window; en lugar de imponer una interfaz de usuario determinada, permite a cada usuario escoger la que más se adapte a sus necesidades.

Es importante mencionar que las terminales gráficas son una evolución de las interfaces de modo texto; no solo son visualmente más agradables, también son muy sencillas de utilizar logrando ser tan accesibles y rápidas como las otras terminales, y esto aunado con las capacidades de Linux, se tiene la posibilidad de emular terminales gráficas sobre un único monitor para lograr mantener sesiones de trabajo distintas.

1.3.3 Terminales virtuales

En Linux existe una emulación de varias terminales sobre un único monitor, estas se denominan frecuentemente terminales virtuales. Para cambiar de una terminal virtual a otra se teclea <Alt><F1>, <Alt><F2>, <Alt><F3>, etc. Con ello se tiene acceso a las terminales /dev/tty1, /dev/tty2, /dev/tty3, etc respectivamente. En cada una de estas terminales puede mantenerse una sesión de trabajo distinta.

Es como si se tuvieran varias terminales físicas formadas por monitor, teclado, memoria, y cable de conexión a la computadora, pero solo usando uno de ellos en un momento dado.

1.3.4 Clientes delgados (Thin Client)

Thin Client es un sistema que consiste de múltiples terminales sin disco (diskless) que están conectadas a un servidor central, las cuales son computadoras configuradas para el trabajo basado en la utilización de recursos remotos a nivel local; es decir, la utilización de los elementos de hardware en el servidor como los discos duros, las unidades de CD-ROM, memoria, etc; así como también del software, que significa poder ejecutar aplicaciones del servidor desde la misma terminal.

En la actualidad comúnmente se hace referencia a un cliente delgado, como una computadora de poca capacidad la cual puede prescindir de un disco duro u otro sistema de almacenamiento secundario y obtener su sistema operativo vía red. Este tipo de equipos pueden conseguirse fácilmente y a un precio muy económico, es muy frecuente que se utilicen computadoras con procesadores 486 o Pentium; ya que éstas se ajustan perfectamente a los requerimientos necesarios de un cliente delgado.

1.3.4.1 Características y Ventajas

- Sistema operativo Abierto
- Booteo remoto
- Alta seguridad
- Costo total de propiedad de software (TCO) reducido
- Administración centralizada
- Eliminación de componentes: unidades de disco, disquete, CD-ROM, etc
- Posibilidad de utilizar equipos en desuso que ya no son aptos para ejecutar las ultimas versiones de los sistemas operativos o diversas aplicaciones

1.3.4.2 Ahorro en costos de software

- Reduce la cantidad de licencias
- Reduce la compra de antivirus
- Herramientas de control y acceso remoto

1.3.4.3 Ahorro en costos indirectos

- Soporte técnico
- Actualizaciones
- Mantenimiento de hardware
- Una terminal con problemas se remplaza rápidamente sin ocasionar perdidas de horas laborables

1.4 Arranque vía red

El desarrollo de la tecnología, la informática y la necesidad actual de conexiones remotas, así como de movilidad y fiabilidad en la gestión de la información han generado una creciente ola de proyectos que tratan de abarcar los requerimientos necesarios para crear soluciones basadas en equipos de cómputo comunes y de poca capacidad.

Estos proyectos aplican la tecnología de clientes delgados para asegurar que se aprovechen los recursos disponibles, los cuales por lo regular cuentan con capacidades suficientes para dar soporte a las necesidades básicas de los usuarios comunes. Para ello es necesario conocer algunas características de los proyectos que ofrecen soluciones basadas en software libre y que además permiten implementar una arquitectura cliente-servidor, como lo es una red de terminales gráficas.

Configurar una terminal gráfica o en general un cliente delgado, implica que se debe cargar el sistema operativo vía red; ésta idea básicamente consiste en que la computadora ejecuta una rutina de arranque, por ejemplo en una ROM o en un disquete, lo que le permite comunicarse con el servidor y obtener una imagen del sistema operativo, el sistema de archivos y ciertas configuraciones necesarias para poder funcionar correctamente.

1.4.1 Etherboot

Es un paquete de software que permite crear imágenes ROM que pueden descargar código a través de una red Ethernet para que éste sea ejecutado en una computadora x86. Muchas tarjetas de red tienen un zócalo donde puede instalarse una ROM y Etherboot es código que puede grabarse en ella. Normalmente este software es usado para implementar PC's diskless y se puede aplicar en varias situaciones, por ejemplo:

- Terminal X
- Clusters
- Ruteadores

Etherboot puede iniciar las computadoras más rápidamente que un disco, debido a que el acceso a un dispositivo mecánico es mucho más lento; incluso con Ethernet 10Mbit, enviar un núcleo de 500Kb tomará solamente un par de segundos; con Ethernet 100Mbit se consigue un mejor resultado.

Se puede trabajar con los RAM Disk, los fileSystems de NFS; o incluso los discos locales si es que así se requiere. Esta es una tecnología que puede ser combinada con otras de manera que se pueda dar solución a un problema muy específico.

Se utiliza principalmente para cargar Linux, FreeBSD o MSDOS; no obstante los protocolos y formatos de archivos de arranque son generales, y no hay razón por la que no podría ser usada para cargar imágenes de otros sistemas operativos.

1.4.2 Netboot

Netboot permite a una computadora con un procesador compatible de Intel, arrancar sin el acceso a un disco duro o a un disquete. Para esto la computadora tiene que ser equipada de modo que pueda cargar el sistema operativo desde un servidor por medio de la red.

Básicamente Netboot puede ser utilizado en:

- Servidor de terminales
- Terminales X

De igual forma que Etherboot, actualmente Netboot puede arrancar Linux, Free BSD y MSDOS.

1.5 LTSP

Este proyecto nació a raíz de la necesidad de resolver un problema para un cliente que necesitaba una terminal que pudiera comunicarse tanto con una IBM AS/400 como con un servidor de aplicaciones Unix. Esta terminal requería TCP/IP, tenía que ser económica así como también fácil de mantener. Además de esto, tenía que permitir al usuario navegar por la Web, enviando y recibiendo e-mail.

Se pudo haber utilizado PC's corriendo Windows, el software estaba disponible, pero el costo hubiera sido muy alto tanto en términos de inversión inicial, así como en soporte y mantenimiento de las PC's a través del tiempo.

Así que se decidió utilizar una computadora sin disco duro, corriendo el kernel Linux y X Window. Basándose en los paquetes Etherboot y Netboot y mediante una económica tarjeta de red que tenía un zócalo para EPROM; inicialmente se probó que en una PC 486 ésta solución funcionó bastante bien. El Linux Terminal Server Project provee una manera simple de utilizar terminales de bajo costo tanto gráficas como terminales de caracteres sobre un servidor GNU/Linux.[8]

Durante la fase de inicio, la terminal obtiene su dirección IP y un kernel desde el servidor, montando luego su sistema de archivos raíz desde el mismo servidor vía NFS.

[8] Pág. web del Proyecto LTSP, Sección de Documentación http://www.ltsp.org/documentation/ltsp-3.0-4-es.html

La terminal puede ser configurada de tres maneras:

- Terminal X: Utilizando X Window, la estación de trabajo puede ser usada para acceder a cualquier aplicación en el servidor, o en cualquier otro servidor dentro de la red.

- Sesiones de telnet basadas en caracteres: Cada estación de trabajo puede invocar múltiples sesiones de telnet al servidor. Cada una de estas sesiones aparecerá en una pantalla virtual separada. Presionando desde Alt-F1 hasta Alt-F9 se cambiará entre cada una de dichas sesiones.

- Prompt de Shell: La estación de trabajo puede ser configurada para iniciar una sesión de Bash en la consola. Esto es muy útil para depurar problemas con X Window o con NFS.

El LTSP es un proyecto de GNU/Linux que permite conectar clientes delgados o terminales a un servidor; la potencia del servidor es extendida a los clientes conectados ya que comparte sus recursos para lograr que estos funcionen en apariencia como computadoras con mayor capacidad.

1.5.1 Características del LTSP

A pesar de que en la actualidad existen varios proyectos similares al LTSP, éste posee ciertas características que le permiten situarse como una de las mejores alternativas para la implementación de redes de terminales gráficas de bajo costo.

A continuación se mencionan las ventajas y desventajas del LTSP, las cuales dan un panorama general de lo que se puede y no se puede hacer con este proyecto.

1.5.1.1 Ventajas

- Uso de software libre, por el cual no se requiere pagar licencias.
- Instalación y configuración de paquetes solo en el servidor.
- Actualización e instalación sencilla de nuevos programas.
- Ejecución de cualquier aplicación GNU/Linux.
- Interfaz gráfica utilizando el X Window System.
- OpenOffice como procesador de texto y hoja de cálculo.
- Mozilla/Opera/Konqueror como navegador web.
- Clientes de e-mail gráficos como Evolution o Kmail.
- Soporte para las distribuciones SuSE, Mandrake, Debian, Fedora Core.
- Optimización en la administración de los servicios y los archivos de los usuarios.
- Protección del sistema al ataque de virus informáticos.

- Disminución de costos en el equipo de cómputo.
- Disminución en los costos de mantenimiento (software y hardware).
- Disminución en los costos de energía debido a que las terminales consumen casi la mitad de energía que una PC normal.

1.5.1.2 Desventajas

- Algunas aplicaciones propietarias no pueden ser ejecutadas en un ambiente LTSP, ya que fueron diseñadas solo para correr en ambientes Windows.
- Ciertas aplicaciones propietarias requieren de muchos recursos para poder ser ejecutadas, por lo que no funcionan adecuadamente de forma remota.

Ya que se ha analizado el concepto del software libre, el origen y la evolución de Linux podemos contar con un panorama que nos servirá para comprender cómo utilizar dicho proyecto, a continuación analizaremos conceptos y algunas definiciones de los componentes de redes así como aspectos técnicos que nos serán útiles en el desarrollo de éste trabajo de tesis.

CONCEPTOS GENERALES

OBJETIVO ESPECÍFICO

Proporcionar los conocimientos básicos mediante conceptos, definiciones y aspectos técnicos, que son requeridos previamente y durante el desarrollo e implementación del LTSP.

2

INTRODUCCIÓN

Conforme las computadoras se van acoplando a la vida diaria, cada vez se usan más para resolver los problemas, y si una sola computadora resulta muy valiosa por su capacidad para procesar información sin necesidad de influencia externa, cómo sería si dicha computadora logrará comunicarse con otras, imagine las capacidades adicionales que estarían disponibles si al conectar las computadoras lograrán desplegar la información contenida en otras máquinas. Considere los beneficios que se lograrían si un grupo de trabajo pudiera acceder al mismo tiempo a un archivo.

Una red hace posible esto y mucho más, es por eso que es importante conocer los diferentes componentes de una red, y que mejor si se pueden encontrar opciones que optimicen el rendimiento de las máquinas tomando en cuenta que no se tenga que invertir mucho.

La instalación de una red no tiene por qué suponer un gran desembolso de dinero para adquirir las computadoras cliente, podemos aprovechar los viejos i486/66 como terminales gráficos. Sólo será necesario que dispongan de tarjeta de red, pudiendo prescindir de unidades de disco duro, disqueteras o cd-rom. Computadoras mejor equipadas no tienen por qué mostrar necesariamente un mayor rendimiento, teniendo en cuenta que las aplicaciones son ejecutadas en el servidor. Los requerimientos del servidor dependerán del número de terminales que atenderá, siendo importante no sólo el procesador sino también la memoria RAM instalada en el servidor.

Por otro lado, el pronosticar a cuántas terminales puede crecer una red es una tarea necesaria para determinar si la alternativa propuesta puede ser viable, o a corto plazo será inoperable. Inicialmente puede escogerse una red de alto rendimiento, pero esto implica un costo más alto, tanto en la instalación como en su mantenimiento.

2.1 Redes

Las redes locales son una alternativa cada vez más viable para medianas y grandes organizaciones es por eso que se necesita dar un panorama para entender como funcionan y así elegir la red que éste de acuerdo a las necesidades y recursos de cada organización.

2.1.1 La red de área local

La red de área local nació con la necesidad de conectar las PC's a fin de compartir información, éstas se pueden clasificar de acuerdo a su distribución geográfica o por el servicio que ofrezcan. Cuando los dispositivos de una red se encuentran en un área mínima como un edificio o una escuela se está hablando de una red de área local, éste sistema de comunicación de alta velocidad permite que las PC's que se encuentran cerca puedan comunicarse entre sí. A su vez la red consta de hardware y software de red y sirve para interconectar las PC's que están aisladas. Esta conexión puede ser directa, por cable coaxial, indirecta o por módem, entre otros. Los diferentes dispositivos que la componen se comunican por una serie de reglas que se conocen como protocolo.

Aunque apareció desde 1983, la LAN ha continuado evolucionando hasta llegar a ser una parte integral de la conectividad de las PC's, además de ser confiable.

El proceso de incorporar una PC a una LAN consiste en la instalación de una tarjeta de interfaz de red (NIC) en cada computadora y esta se conecta con un cable especial para red. El último paso para implantar una LAN es instalar en cada PC un software conocido como sistema operativo de red (NOS).

Si se dividiera una red en sus componentes más simples, se tendrían dos partes, la física: el cableado, las tarjetas de red, las computadoras y demás equipo que utiliza la red para transmitir los datos. La otra parte es la disposición lógica de esos componentes físicos: las reglas que permiten a los componentes físicos trabajar en conjunto. En general a estas dos partes se les conoce como hardware y software de red.

2.2 Hardware de red

Las redes están conformadas por muchos elementos de hardware diferentes que deben trabajar juntos para crear una red funcional, estos dispositivos los fabrican por lo general varias compañías, por lo tanto es necesario que haya comunicación entre los fabricantes en relación con la manera en que los componentes de la red trabajan e interactúan entre sí, esto con el fin de lograr una mayor compatibilidad entre ellos. La conectividad del hardware de red depende de las características con las que fue diseñado y también de la manera en que se implementa; ya que existen múltiples variantes en cuanto a la instalación y configuración de los diferentes elementos de la red.

2.2.1 Topologías

Los nodos de red necesitan estar conectados para comunicarse y a la forma en que están conectados se le llama topología, que consta de una topología física y una lógica. La física se refiere a la manera en que los nodos están conectados unos con otros y la lógica es el método que se usa para comunicarse con los demás nodos, es decir; la ruta que toman los datos dentro de la red para llegar a los diferentes nodos. Las topologías física y lógica pueden ser iguales o diferentes.

Cada red local usa un cable para llevar la información, este cable debe controlar el flujo de la información en la red de tal forma que los mensajes puedan ser transferidos de una manera confiable. En la mayoría de las topologías las señales son enviadas en todas direcciones desde el equipo emisor, cada dispositivo tiene su propia dirección asignada, y el software se utiliza para programar un dispositivo que acepte mensajes con dirección única, e ignora a todos los demás.

Las 3 topologías de red estándar son de bus, estrella y anillo; también hay combinaciones de más de una topología.

2.2.1.1 Topología de bus

En una topología de bus, cada computadora está conectada a un segmento común de cable de red. El segmento de red se coloca como un bus lineal, es decir, un cable largo que va de un extremo a otro de la red, y al cual se conecta a cada nodo. El cable puede ir por el piso, por las paredes, por el techo, o puede ser una combinación de éstos, siempre y cuando el cable sea un segmento continuo.

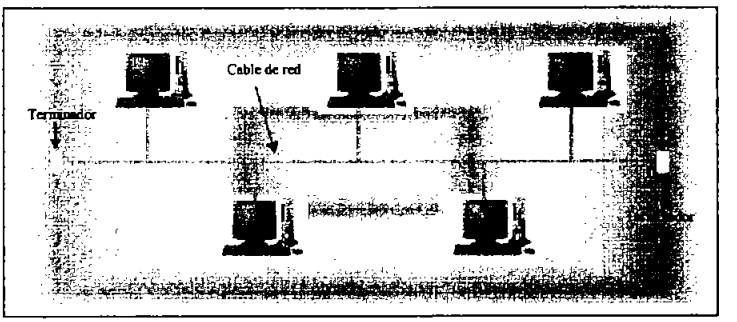


Figura 2-1 Topología de bus

2.2.1.2 Topología de estrella

En una topología de estrella, cada computadora está conectada a un concentrador (o hub) ubicado centralmente. El concentrador es un dispositivo de hardware con varios puertos, al cual se puede conectar un cable de red en cada uno de ellos, donde cada máquina tiene un enlace exclusivo con el hub.

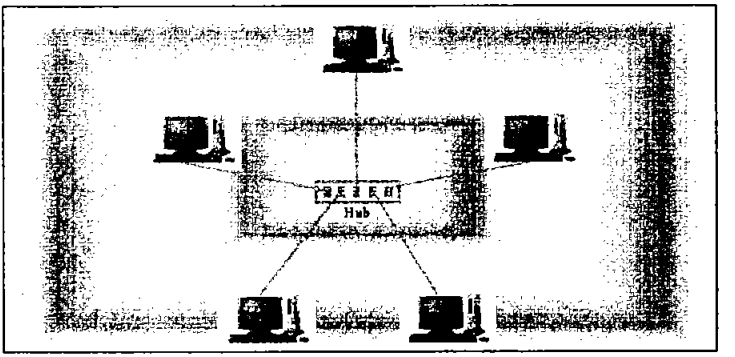


Figura 2-2 Topología de estrella

2.2.1.3 Topología de anillo

En una topología de anillo, cada computadora se conecta en forma circular a la red. Las topologías de anillo casi siempre son lógicas con topología física de estrella. La topología física muestra que cada computadora se conecta a un dispositivo central y parece una estrella. La ruta seguida por los datos de una computadora a otra muestra que la topología lógica es de anillo.

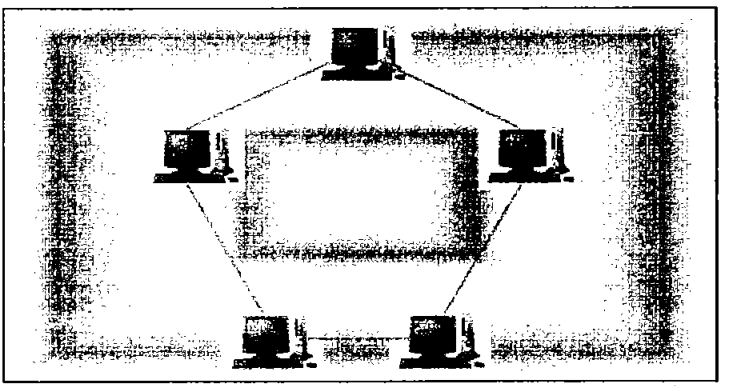


Figura 2-3 Topología de anillo

2.2.1.4 Topología de árbol

Una topología de árbol es la combinación de las topologías de bus y de estrella. Muchos concentradores de las redes Ethernet con topología física de estrella también tiene un conector en la parte trasera que enlaza al concentrador a una red de topología física de bus.

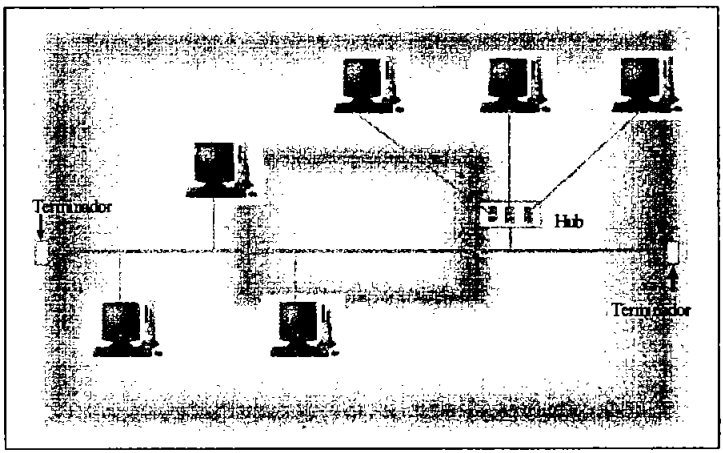


Figura 2-4 Topología de árbol

2.2.2 Estándares de redes

Por suerte se han creado estándares que definen la forma de conectar componentes de hardware en las redes y el protocolo (o reglas) de uso cuando se establecen comunicaciones por red. Donde el protocolo define la forma en que deben efectuarse las comunicaciones de las redes.

Un estándar es la especificación de red adoptada, incluye reglas que se refieren al tipo de componentes que deben usarse, a la manera de conectar los componentes, así como a los protocolos de comunicación que hay que emplear.

Los 3 estándares más populares que se utilizan son:

- Ethernet
- ARCnet
- Token Ring.

2.2.2.1 Ethernet

Ethernet también se le conoce como IEEE 802.3, es el estándar más popular para las LAN que se usa actualmente y transmite datos a través de la red a una velocidad de 10 Mbps (megabits por segundo). Este estándar usa un método de acceso al medio conocido como Acceso Múltiple con Detección de Portadora y Detección de Colisiones (CSMA / CD). Antes de que un nodo envié algún dato a través de la red, primero escucha y detecta si algún otro está transfiriendo información.

De no ser así, el nodo transfiere la información a través de la red. Los otros nodos escucharán y el nodo seleccionado recibirá la información. En caso de que dos nodos traten de enviar datos por la red al mismo tiempo, cada nodo se dará cuenta de la colisión y esperará una cantidad de tiempo aleatoria antes de volver a hacer el envío.

Aunque CSMA / CD es una forma rápida y eficiente para transmitir datos, una red muy cargada podría llegar al punto de saturación. Conforme más nodos tratan de transmitir información por la red, más aumenta la posibilidad de colisiones y se reduce de modo importante la eficiencia de la red. Sin embargo, con una red diseñada adecuadamente, la saturación rara vez es preocupante.

Existen 3 estándares de Ethernet, 10BASE5, 10BASE2 y 10BASE-T, que definen el tipo de cable de red, las especificaciones de longitud y la topología física que debe utilizarse para conectar nodos en la red.

2.2.2.1.1 10BASE5

También llamado Ethernet estándar, Thick Ethernet o Thicnet, tiene un estándar de topología física de bus que consiste en un segmento de cable de red con terminadores en los extremos. Los terminadores incluyen una resistencia que disipa la señal de la red y no permite que se refleje de regreso al cable de red. Su NIC está conectada a un transmisor-receptor (transceptor) externo por medio de un cable de suspensión. El transceptor está conectado al segmento de cable Thicnet y actúa para transmitir y recibir datos de la red entre la computadora y la red.

Es relativamente difícil trabajar con Thicnet, en comparación con las otras dos disposiciones, 10BASE2 y 10BASE-T. Sin embargo debido a que fue la única disponible durante un tiempo, Thicnet se encuentra todavía en varias instalaciones.

2.2.2.1.2 10BASE2

También llamado Thinnet, Thin coax o Thin Ethernet, éste se instala por medio de una topología física de bus, que consiste en segmentos de cable con terminaciones en

cada extremo. La NIC de cada computadora está conectada directamente al segmento de cable Thinnet; lo que descarta la necesidad de un transceptor externo.

Thinnet fue muy popular en negocios e instalaciones pequeños, debido a que es el método menos caro para poner en servicio una red Ethernet ya que se emplea una pequeña cantidad de nodos. Además es menos susceptible a la interferencia eléctrica que el par trenzado. Una desventaja, es que si hay una ruptura en cualquier parte del cable, dejará de funcionar toda la red. Así mismo puede ser ardua la búsqueda de fallas causadas por un problema de cable.

2.2.2.1.3 El 10BASE-T

Éste a diferencia del Thick o Thin Ethernet, se instala por medio de una topología física de estrella donde cada nodo se conecta a un hub y la NIC de cada computadora se conecta al concentrador por medio de un segmento de cable de red.

Con frecuencia se prefiere el 10BASE-T sobre el 10BASE2, por su flexible topología de estrella; ya que la ruptura en el cable de una red con 10BASE-T solo desactivará a la computadora que esté al extremo de la línea rota, en vez de toda la red como sucede con el otro estándar. El 10BASE-T es más barato para redes pequeñas aunque requiere un concentrador adicional; sin embargo el cable de par trenzado que se emplea en 10BASE-T es menos caro, por lo que entre más nodos se añadan, el gasto adicional de un concentrador será menor en comparación con el costo de utilizar el cable Thinnet.

2.2.2.2 Token Ring

Llamado IEEE 802.5, este estándar opera a una velocidad de 4 Mbps o 16 Mbps, emplea topología lógica de anillo y una topología física de estrella. La NIC de cada computadora se conecta a un cable y éste a su vez a un hub llamado unidad de acceso a multiestaciones (MAU). Token Ring se basa en un esquema de paso de señales (token passing) es decir, se transmite una señal a todas las computadoras de la red donde la computadora que esté en posesión del token tiene autorización para transmitir su información a otro nodo.

Cuando termina, el token pasa a la siguiente computadora, si ésta tiene que enviar información acepta el token y procede a enviarla; en caso contrario, el token pasa a la siguiente computadora del anillo y el proceso continúa.

La MAU se salta automáticamente un nodo de red que no esté encendido. Sin embargo, dado que cada nodo de una red Token Ring examina y luego retransmite cada token, un nodo con mal funcionamiento puede hacer que deje de trabajar toda la red.

2.2.2.3 ARCnet

ARCnet es un estándar similar al IEEE 802.4 y en 1992 ANSI lo reconoció como estándar formal. Soporta una velocidad de transferencia de datos de 2.5 Mbps, usa una topología lógica de bus y una ligera variación de la topología física de estrella.

Cada nodo está conectado a un concentrador pasivo o a uno activo. Un concentrador pasivo no recibe potencia eléctrica y sirve para distribuir la señal de la red a cortas distancias, en cambio un concentrador activo sí recibe potencia eléctrica y también amplifica la señal de la red para permitir que la red cubra distancias más largas.

La NIC en cada computadora está conectada a un cable que a su vez esta conectado a un concentrador activo o pasivo.

2.2.3 Adaptadores de red

Para comunicarse con el resto de la red, cada computadora debe tener instalada una tarjeta de interfaz de red (NIC), la cual se conecta a otros nodos por medio del cable de red empleando y los protocolos adecuados.

Aunque la mayor parte de los adaptadores de red se instalan en el interior de la computadora, algunos son de instalación externa. El tipo de adaptador de red que se compre determinará la velocidad de transmisión de datos, el tamaño de las unidades de mensaje (paquetes), así como la topología de la red, por lo que hay que estar seguro de obtener el adaptador de red adecuado para la topología que se quiera usar.

Para que un adaptador funcione como interfaz entre la computadora y la red, debe satisfacer 2 criterios:

1. Debe ser del tipo adecuado para la red con la que se va a conectar, utilizando los protocolos correspondientes para comunicarse con el resto de la red y tener el conector apropiado para el cable.

2. Debe tener el conector apropiado para la ranura de expansión de la computadora ya que en ocasiones los equipos de diferentes fabricantes tienen distintos tipos de ranuras de expansión.

La mayor parte de las computadoras tienen ranuras de expansión ISA, las cuales son de 8 o 16 bits. Actualmente las computadoras cuentan con ranuras de expansión, como el bus local VESA o el bus local PCI que permiten que pasen los datos en la ranura de bus local a mayor velocidad que las ranuras ISA estándar.

2.2.3.1 Adaptadores Ethernet

En la actualidad, casi todas las redes se basan en el estándar Ethernet y por lo tanto lo más probable es que cuenten con tarjetas de red Ethernet.

2.2.3.1.1 10BASE5

Tiene un conector DB-15 hembra tipo D de 15 pines, el cable que se utiliza es el Thin Ethernet que se acopla al resto de la red a través de un transmisor-receptor externo.

2.2.3.1.2 10BASE2

Tiene un conector BNC que se une al cable por medio de un T BNC y las otras dos partes de la T están conectadas al cable Thinnet que va hacia otros nodos de la red. Aunque Thinnet tiene topología física de bus, en realidad el adaptador de red se bifurca y se dirige hacia el segmento de cable de red mediante el conector T BNC, por lo que si se retira el conector de la tarjeta no afectará la conexión de la red.

2.2.3.1.3 10BASE-T

Los adaptadores de red que soportan 10BASE-T usan un conector RJ-45, similar al conector RJ-11 de la mayor parte de las instalaciones telefónicas, a excepción de que el conector RJ-45 es más grande y tiene ocho conductores en vez de cuatro. El cable de red UTP tiene una clavija RJ-45 de la tarjeta de red y el otro extremo al socket RJ-45 del concentrador, los otros nodos de la red se conectan al concentrador en forma similar.

2.2.3.2 Dirección MAC

Las tarjetas de red tipo Ethernet tienen una pequeña memoria en la que alojan un dato único para cada tarjeta de este tipo. Se trata de la dirección MAC, y está formada por 48 bits que se suelen representar mediante dígitos hexadecimales que se agrupan en seis pares (cada par se separa de otro mediante dos puntos ":" o mediante guiones "-").

Por ejemplo, una dirección MAC podría ser F0:E1:D2:C3:B4:A5.

La mitad de los bits de la dirección MAC son usados para identificar al fabricante de la tarjeta, y los otros 24 bits son utilizados para diferenciar cada una de las tarjetas producidas por ese fabricante.

2.2.4 Cables para red

En este punto se debe decidir que tipo de cable se utilizará para conectar las terminales y el servidor. Debido a que la instalación y la elección del cable constituyen generalmente el cincuenta por ciento de la instalación completa, se deben planear cuidadosamente estos aspectos.

Cada estándar de red define el tipo de cableado que se requiere y las especificaciones para la conexión de los nodos de la red. A causa de las altas velocidades y a la gran cantidad de datos que se transmiten a través del cable, las especificaciones del mismo y las reglas para su uso son muy estrictas. El uso de cable con especificaciones incorrectas causa a final de cuentas, fallas en las comunicaciones de la red.

Los estándares especificados para la longitud del cable se refieren por lo general a segmentos de cable. Un segmento de cable es una sección continua que no es interrumpida por ningún dispositivo, como un concentrador o un repetidor. Se pueden incluir conectores en un segmento de cable de red, a causa de que sencillamente se conectan en el segmento sin interrumpir o cambiar la señal de la red.

Existen compañías especializadas en cables de datos y de telecomunicaciones que pueden configurar cables de longitud a la medida, de acuerdo con requerimientos particulares.

Si se tiene que instalar un vasto cableado o si se tienen requerimientos de cable especiales, se puede comprar el cable a granel e instalar uno mismo los conectores. Si se escoge este último método, es indispensable que se obtengan las herramientas y los conectores adecuados.

2.2.4.1 Ethernet

Hay cuatro tipos de cable para una red Ethernet: Thick Ethernet (10BASE5), Thin Ethernet (10BASE2), UTP (10BASE-T) y fibra óptica (10BASE-F). Los cables de red utilizados para cada tipo de Ethernet no pueden mezclarse, aunque existen dispositivos que permiten la conexión de diferentes tipos de segmentos de red Ethernet.

2.2.4.1.1 Thick Ethernet

El cable empleado por Thick Ethernet (Ethernet gruesa) es un tipo especial de cable coaxial. El conductor central está rodeado por un aislante dieléctrico al que a su vez, lo rodea un blindaje de hoja de metal. Alrededor del blindaje, hay un conductor tejido rodeado por otro blindaje de hoja de metal que, también está cubierto por un conductor tejido. La parte externa del cable tiene una cubierta protectora.

La reglas para la instalación y configuración de segmentos de cable Thick Ethernet son las siguientes:

- La longitud máxima de segmento de red es de 500 metros.
- Cada segmento de red debe tener una terminación de 50 ohms (ohmios) en cada extremo.
- No pueden conectarse en serie más de cinco segmentos de red y solo tres de éstos pueden estar ocupados (tener nodos conectados a ellos).
- La cantidad máxima de transmisores-receptores por segmento es de 100.
- La cantidad máxima de nodos en una red es de 1024.
- Los transceptores no pueden instalarse a menos de 2.5 metros.
- Los cables de bajada no pueden ser más largos de 50 metros.

A los conectores para el cable Thick Ethernet (Ethernet gruesa) se les llama conectores coaxiales serie N.

2.2.4.1.2 Thin Ethernet

El cable Thin Ethernet (Ethernet delgada), es un tipo de cable coaxial RG-58 que consiste en un conductor interno rodeado por un aislante dieléctrico, un blindaje de hoja de metal, un conductor tejido y una cubierta exterior protectora.

Para la instalación y configuración del cable Thin Ethernet, se aplican las siguientes reglas:

- La longitud máxima del segmento debe ser de 185 metros.
- Cada segmento de red debe tener una terminación de 50 ohms (ohmios) en cada extremo.
- No pueden conectarse en serie más de cinco segmentos de red y solo tres de éstos pueden estar ocupados (tener nodos conectados a ellos).
- La cantidad máxima de nodos por segmento es 30.
- La distancia mínima de cable entre adaptadores de red es 0.5 metros.
- La cantidad máxima de nodos en una red es de 1024.

Se usan conectores tipo BNC para el Thin Ethernet.

2.2.4.1.3 UTP Ethernet

El UTP es un cable que consiste en pares trenzados entre ellos, Ethernet UTP emplea un total de cuatro conductores (o dos pares) para transmitir y recibir la señal de red. Puesto que los conectores estándar RJ-45 tienen ocho números de conexión, el cable

que se instala tiene generalmente ocho conductores aunque la red solo use cuatro de ellos.

Adicionalmente al cableado de par trenzado viene en varios niveles que van desde el nivel 1 (o Categoría 1) utilizado con frecuencia para aplicaciones telefónicas, hasta nivel 5 (o Categoría 5) que está certificado para la transmisión de datos a una velocidad de hasta 155 megabits por segundo.

Para la instalación y configuración del cable Ethernet UTP, se aplican las siguientes reglas:

- La longitud máxima de cable entre un nodo y un concentrador es de 100 metros.
- La especificación IEEE para Ethernet 10Base-T requiere usar solo dos pares trenzados los pines 1, 2, 3 y 6 del conector RJ-45 son conectados de manera directa, donde 1 y 2 son transmisores y el 3 y 6 receptores.
- Se pueden conectar hasta 12 concentradores a un concentrador principal.
- Sin el uso de puentes, el cable Ethernet UTP puede acomodar un máximo de 1024 estaciones de trabajo.

Es importante conocer las normas para la configuración de los pines de un cable de red, ya que en algunos casos se necesita construir un cable directo y en otros un cable cruzado.

De acuerdo con la Norma EIA/TIA 568A RJ45, el orden de los pines es el siguiente:

Pin 1	Verde/Blanco
Pin 2	Verde
Pin 3	Naranja/Blanco
Pin 4	Azul
Pin 5	Azul/Blanco
Pin 6	Naranja
Pin 7	Café/Blanco
Pin 8	Café

Tabla 2-1 Norma EIA/TIA 568A RJ45

Y la Norma EIA/TIA 568B RJ45, especifica el siguiente orden:

Pin 1	Naranja/Blanco
Pin 2	Naranja
Pin 3	Verde/Blanco
Pin 4	Azul
Pin 5	Azul/Blanco
Pin 6	Verde
Pin 7	Café/Blanco
Pin 8	Café

Tabla 2-2 Norma EIA/TIA 568B RJ45

Para conectar una PC a un concentrador o a un switch se necesita un cable directo, este cable tendrá en ambos extremos la misma norma, pero si lo que se requiere es conectar 2 dispositivos del mismo tipo, esto se hace por medio de un cable cruzado; este cable se caracteriza porque en sus extremos se utilizan diferentes normas.

2.2.4.1.4 Fibra óptica

El cable de fibra óptica transmite datos por medio de una serie de pulsos de luz, transmitidos a través de una hebra fina de vidrio. Es capaz de transmitir datos a alta velocidad y no es susceptible de interferencias exteriores, la instalación requiere de cuidados especiales ya que sus empalmes y conectores son más costosos y difíciles de manejar. Actualmente se dispone de adaptadores de red que se conectan directamente a la fibra, ésta se emplea, por lo general, para conetar redes que requieren comunicación de alta velocidad en distancias mucho más largas.

2.2.5 Dispositivos de expansión

El consumidor promedio de una red de área local generalmente busca una solución rentable a sus problemas inmediatos, también desea incrementar la calidad y hacer más fácil su trabajo. Sin embargo, no todas las redes de área local fueron diseñadas teniendo como consideración principal el rendimiento.

Uno de los puntos que se deben tomar en cuenta al escoger una red, son la disposición física de la misma y los requerimientos de carga de trabajo previstos, de manera que se pueda determinar el posible rendimiento de la misma.

Si se necesita cubrir una distancia mayor se deberá escoger un hardware de red que lo permita. En el laboratorio de electrónica se cuenta con un switch como dispositivo de expansión, pero existen otros que pueden extender la longitud de la red. A continuación se describen los dispositivos usados por Ethernet.

Cada uno de los dispositivos y de los métodos usados para expandir la red tiene un propósito específico. Sin embargo muchos de ellos incorporan las características de otros tipos de dispositivos para aumentar su flexibilidad.

2.2.5.1 Repetidor

Un repetidor es un dispositivo que permite extender la longitud o cobertura de una red ya que amplifica y retransmite las señales, toma la señal que circula por un segmento de una red homogénea y la reenvía sin efectuar ningún tipo de interpretación; es capaz de conectar diferentes medios físicos de transmisión, sin embargo no suele utilizarse para conectar redes de banda base con redes de banda ancha ya que los métodos de decodificación de la información son muy diferentes.

2.2.5.2 Hub

Los hubs o concentradores son equipos que permiten compartir el uso de una línea o punto de red entre varias computadoras, todas ellas pueden usar la línea aunque no de forma simultánea, ni utilizando distintos protocolos, ni distintas velocidades de transmisión.

Los hubs funcionan repitiendo cada paquete de datos en cada uno de los puertos con los que cuenta de forma que todos los puntos tienen acceso a los datos.

Estos dispositivos son un punto central de conexión para nodos de red que están dispuestos de acuerdo a una topología física de estrella, como en el Ethernet 10BASE-T.

2.2.5.3 Switch

Los conmutadores (switch ethernet) se caracterizan por no enviar los paquetes a todos los puertos, sino únicamente al puerto correspondiente del destinatario. La diferencia entre un conmutador y un puente es que el puente debe recibir todo el paquete antes de dirigirlo al puerto correspondiente y un conmutador dirige el paquete a su destino una vez recibido el encabezado del paquete (en ella se encuentra la dirección IP del destinatario); gracias a ello, los conmutadores producen un retraso mínimo en la conmutación.

De esta manera, utilizando un conmutador se puede dividir una red en varios segmentos y limitar el tráfico al segmento o segmentos a los que pertenece el paquete. Su utilización permite que cada usuario o grupo de usuarios tenga su propio segmento y ancho de banda dedicados con mucha menor tasa de colisiones y un menor tiempo de respuesta en lugar de lo que ocurre en una red Ethernet tradicional en la que muchos usuarios compartan el mismo ancho de banda.

2.2.5.4 Puente

Un puente es un dispositivo que conecta dos redes LAN separadas para crear lo que aparenta ser una sola. Los puentes revisan la dirección asociada con cada paquete de información, si la dirección es la correspondiente al otro segmento de red, el puente pasará el paquete al segmento; si el puente reconoce que la dirección es la correspondiente a un nodo del segmento actual, no dejará pasar el paquete al otro lado.

Los puentes también suelen emplearse para reducir la cantidad de tráfico en una red, mediante la división de un solo segmento de red en dos y conectándolos por medio de éste.

2.2.5.5 Ruteador

Los ruteadores son similares a los puentes, sólo que operan a un nivel diferente, estos requieren por lo general que cada red tenga el mismo NOS. Con un NOS común, el ruteador puede ejecutar funciones más avanzadas de las que podría permitir un puente, como conectar redes basadas en topologías lógicas completamente diferentes como Ethernet y Token Ring. Los ruteadores también suelen ser lo suficientemente inteligentes para determinar la ruta más eficiente para el envío de datos, en caso de haber más de una ruta. Sin embargo, junto con la complejidad y la capacidad adicionales proporcionadas por los ruteadores se da una penalización de aumento de costo y un rendimiento disminuido (los ruteadores cuestan mucho más que los puentes).

2.2.5.6 Compuerta

Una compuerta (gateway) es un sistema formado por hardware y software que permite que los nodos de una red se comuniquen con tipos diferentes de red o con otros dispositivos. Podría tenerse, por ejemplo, una LAN que consista en computadoras compatibles con IBM y otra que consista en computadoras Macintosh. En este caso, una compuerta permitiría que las computadoras IBM compartieran archivos con las Macintosh. Este tipo de compuertas también permite que se compartan impresoras entre las dos redes.

2.3 Software de red

Para poder hacer una selección adecuada del NOS, es importante tener una opinión objetiva de personas que han experimentado, o están de alguna manera involucradas en el funcionamiento de estos sistemas, ya que al haberse enfrentado personalmente con uno o más de ellos, tienen una perspectiva un poco más amplia de sus pros y contras.

El software de red consiste en programas que implementan protocolos o normas de comunicación, para que las computadoras puedan compartir su información; éstos protocolos se aplican enviando y recibiendo grupos de datos formateados denominados paquetes.

Éste software también se encarga de efectuar conexiones lógicas entre las aplicaciones de la red, dirigir el movimiento de paquetes a través de la red física y minimizar las posibilidades de colisión entre paquetes enviados simultáneamente. En general este conjunto de programas forman parte de lo que se conoce como sistema operativo de red.

2.3.1 Network Operative System

El NOS permite que el software de aplicación (el procesador de palabras, las bases de datos, las hojas de cálculo, los paquetes de contabilidad, etc.) que se esté ejecutando en una computadora se comunique a través de la red con otras computadoras. El propósito principal de un sistema operativo es dar el soporte que necesitan los programas que hacen un trabajo de interés para el usuario. Por ejemplo, se puede estar utilizando un editor si se quiere crear un documento, ese editor no puede realizar su trabajo sin la ayuda del sistema operativo, necesitará ésta ayuda para interactuar con la terminal, los archivos y el resto de la computadora. Debido a que existen varios sistemas operativos de red en el mercado, es necesario observar la manera en que éstos están posicionados en el gusto del comprador.

Novell con sus diferentes versiones de NetWare fue el principal protagonista en el campo de las redes desde la década de los 80's. Microsoft tuvo un inicio lento con su sistema Windows NT, pero tuvo un gran auge que lo llevó a estar en segundo sitio detrás de Novell. Con menos fuerza se repartieron lo que restaba del mercado; el sistema OS/2 de IBM, la solución para redes cliente/servidor AppleShare de Apple y las diferentes variantes de Unix como Solaris, UnixWare y Linux.

Al momento de seleccionar el sistema operativo de red para una LAN, se necesita tener información sobre ciertos aspectos, que pueden influir de manera directa o indirecta en el éxito de la implementación del NOS en la red. Algunos de estos requerimientos son: personal capacitado, conocimiento de costos de cada NOS,

requerimientos de hardware y compatibilidad de los sistemas a instalar con otros similares.

Se propone utilizar Linux, que es un sistema operativo multiusuario y multitarea, tiene las mismas características que Unix pero también aporta elementos propios con la gran ventaja de ser software libre. Linux soporta el modo protegido de los procesadores de Intel 80x86, pero también las prestaciones más avanzadas de todos los procesadores que soporta su núcleo: PowerPC, SPARC, MIPS, sistemas IBM 390, PDA's, etc.

El entorno gráfico le aporta a Linux vistosidad por un lado pero facilidad de manejo por otro, al igual que los entornos gráficos de otros sistemas (Microsoft Windows y Apple Mac) X Window ofrece un entorno multiventana. En Linux se pueden ejecutar distintos gestores de ventanas, desde los conocidos Motif y Open Look originarios de Unix, hasta los más conocidos en la actualidad, Gnome y KDE.

Algunas características técnicas de este sistema operativo son:[9]

- Multitarea. Varios programas (realmente procesos) ejecutándose al mismo tiempo.
- Multiusuario. Varios usuarios en la misma máquina al mismo tiempo.
- Multiplataforma. Corre en muchas CPU's distintas, no sólo Intel.
- Tiene protección de la memoria entre procesos, de manera que uno de ellos no pueda detener el sistema.
- Memoria virtual usando paginación (swap).
- Todo el código fuente está disponible, incluyendo el núcleo completo y todos los drivers, las herramientas de desarrollo y todos los programas de usuario.
- Consolas virtuales múltiples. Varias sesiones de login a través de la consola entre las que se puede cambiar con las combinaciones adecuadas de teclas (totalmente independiente del hardware de video). Se crean dinámicamente y puedes tener hasta 64.
- Soporte para varios sistemas de archivo comunes, incluyendo minix-1, Xenix y todos los sistemas de archivo típicos de System V.
- TCP/IP, incluyendo FTP, telnet, NFS, etc.
- Software cliente y servidor Netware disponible en los núcleos de desarrollo.

2.3.2 Protocolos

Muchos componentes del NOS están basados en diversos protocolos, los cuales definen la forma en que se va a comunicar la computadora con otros dispositivos dentro de una red. Se puede definir un protocolo como el conjunto de normas que regulan la comunicación (establecimiento, mantenimiento y cancelación) entre los distintos componentes de una red informática. Existen dos tipos de protocolos: de bajo nivel y de red.

---

[9] FLORIANO Blanco Lino, GALÁN Galán Susana, Linux Al día en una hora, Ed. Anaya Multimedia

Los protocolos de bajo nivel controlan la forma en que las señales se transmiten por el cable o medio físico. Los protocolos de red organizan la información (controles y datos) para su transmisión por el medio físico a través de los protocolos de bajo nivel.

En Linux se tiene implementado el conjunto de protocolos de red TCP/IP, los cuales permiten comunicar e identificar a los diferentes dispositivos dentro de una red por medio de direcciones IP.

2.3.3 Dirección IP

El Protocolo de Internet (IP) y el Protocolo de Transmisión (TCP), fueron desarrollados inicialmente en 1973 por el informático estadounidense Vinton Cerf como parte de un proyecto dirigido por el ingeniero norteamericano Robert Kahn y patrocinado por la Agencia de Programas Avanzados de Investigación (ARPA, siglas en inglés) del Departamento Estadounidense de Defensa.

La especificación original de TCP/IP era abierta y así lo pensaron sus diseñadores, ellos crearon un espacio de direcciones o una manera estándar de escribirlas, la cual establecía 2^{32} direcciones (es decir, 4 294 967 296 direcciones diferentes). Estas direcciones permiten identificar de manera única a una computadora dentro de toda la red de Internet, por lo que se puede enviar información a otra computadora con solo saber su dirección IP; todo el proceso para llevar la información hasta la máquina destino queda en manos de la infraestructura propia de Internet (gateways, ruteadores, puentes, etc.).

Debido a que cada host en Internet requiere de un nombre y dirección únicos, una vez que se agote el espacio actual de direcciones IP de cuatro mil millones de direcciones, no habrá más direcciones. Esta es la razón por la cual es tan importante la siguiente generación del Protocolo de Internet llamada IPv6.

La razón por la cual las direcciones IP se han agotado tan rápido se debe al diseño del esquema de direccionamiento, todas las direcciones IP se escriben en notación decimal punteada, con un byte entre cada punto. Una dirección IP tienen el siguiente formato: X.X.X.X. Debido a que cada número está descrito por un byte, y que cada byte tiene 8 bits; cada número puede tener un valor cualquiera entre 0 y 225. Ya que existen 4 números con 8 bits cada uno, se dice que el espacio total de direcciones tiene una longitud de 32 bits.

Con un espacio de direcciones de 32 bits se podría pensar que nunca se terminarían las direcciones IP; sin embargo debido a la forma en que ha crecido Internet está sucediendo lo contrario, donde las direcciones IP se asignan a organizaciones que las solicitan en lo que se llaman bloques de direcciones. Los bloques de direcciones tienen tres tamaños, basándose en lo que se le conoce como clases de direcciones.

2.3.3.1 Clasificación

Existen 5 clases de direcciones de redes: A, B, C, D o E (esta diferenciación viene dada en función del número de computadoras que va a tener la red).

- Direcciones clase A: Contiene 7 bits para direcciones de red (el primer bit del primer octeto siempre es un 0) y los 24 bits restantes representan a direcciones de equipo. De esta manera, permite tener un máximo de 128 redes (aunque en realidad tiene 126, ya que están reservadas las redes cuya dirección de red empieza por 0 y por 127), cada una de las cuales puede tener 16, 777, 216 computadoras (aunque en realidad tiene 16, 777, 214 computadoras cada una, ya que se reservan aquellas direcciones de equipo en binario, cuyos valores sean todos ceros o todos unos). Las direcciones en representación decimal, estarán comprendidas entre 0.0.0.0 y 127.255.255.255 donde la máscara de subred será de 255.0.0.0.

- Direcciones clase B: Contiene 14 bits para direcciones de red (ya que el valor de los 2 primeros bits del primer octeto son siempre 10) y 16 bits para direcciones de equipo, lo que permite tener un máximo de 16, 384 redes, cada una de las cuales puede tener 65, 536 computadoras (aunque en realidad tiene 65, 534 computadoras cada una, ya que se reservan aquellas direcciones de equipo en binario cuyos valores sean todos ceros o todos unos). Las direcciones en representación decimal, estarán comprendidas entre 128.0.0.0 y 191.255.255.255 donde su máscara de subred será de 255.255.0.0.

- Direcciones clase C: Contiene 21 bits para direcciones de red (ya que el valor de los 3 primeros bits del primer octeto son siempre 110) y 8 bits para direcciones de equipo, lo que permite tener un máximo de 2, 097, 152 redes, cada una de las cuales puede tener 256 computadoras (aunque en realidad tiene 254 computadoras cada una, ya que se reservan aquellas direcciones de equipo en binario cuyos valores sean todos ceros o unos). Las direcciones en representación decimal, estarán comprendidas entre 192.0.0.0 y 223.255.255.255, por lo que su máscara de subred será de 255.255.255.0.

- Direcciones clase D: Se reserva todas las direcciones para multidestino (multicasting), es decir, una computadora transmite un mensaje a un grupo específico de computadoras de esta clase. El valor de los 4 primeros bits del primer octeto son siempre 1110 y los últimos 28 bits representan los grupos multidestino. Las direcciones en representación decimal estarán comprendidas entre 224.0.0.0 y 239.255.255.255.

- Direcciones clase E: Se utiliza con fines experimentales únicamente y no está disponible para el público. El valor de los 4 primeros bits del primer octeto son siempre 1111 y las direcciones en representación decimal estarán comprendidas entre 240.0.0.0 y 255.255.255.255.

2.3.3.2 Máscara de subred

La máscara de subred indica la localización de cada parte de una dirección IP, es decir, cuántos bits corresponden al identificador de la red y subred, además de cuántos se destinan al identificador de host.

Está formada por 32 bits que toman valor 1 ó 0

- Valor 1 para los bits cuya posición corresponde a bits de identificación de red
- Valor 0 para los bits cuya posición corresponde al identificador de host

Para que su manejo sea más sencillo se dividen en octetos, se pasan a decimal, y se separan por puntos. Los bits correspondientes a la identificación de subredes se toman siempre contiguos para obtener un rango continuo de direcciones de host. La configuración de cualquier dispositivo en TCP/IP requiere siempre una dirección IP única y el valor correspondiente a su máscara de subred ya sea genérica o personalizada para crear subredes.

Para cada clase de direcciones hay definidas mascaras genéricas:

- CLASE A: 255.0.0.0
- CLASE B: 255.255.0.0
- CLASE C: 255.255.255.0

A la hora de decidir que máscara de subred se utilizará hay que tener en cuenta la red y su posible crecimiento, es decir el aumento en número de subredes y el aumento en número de equipos. El LTSP particularmente utiliza la máscara de subred de la clase C "255.255.255.0", la cual permitirá tener hasta 254 terminales y un servidor.

2.3.4 Arquitectura cliente - servidor

La arquitectura cliente/servidor es un modelo para el desarrollo de sistemas de información, en el que las transacciones se dividen en procesos independientes que cooperan entre sí para intercambiar información, servicios o recursos. Se denomina cliente al proceso que inicia el diálogo o solicita los recursos, y servidor al proceso que responde a las solicitudes. Es el modelo de interacción más común entre aplicaciones en una red, no forma parte de los conceptos de Internet como los protocolos IP, TCP o UDP, sin embargo todos los servicios estándares de alto nivel propuestos en Internet funcionan según este modelo. Los principales componentes del esquema cliente/servidor son los clientes, los servidores y la infraestructura de comunicaciones.

En este modelo, las aplicaciones se dividen de forma que el servidor contiene la parte que debe ser compartida por varios usuarios, y en el cliente permanece sólo lo particular de cada usuario. Los clientes interactúan con el usuario, usualmente en forma

gráfica, frecuentemente se comunican con procesos auxiliares que se encargan de establecer conexión con el servidor, enviar el pedido, recibir la respuesta, manejar las fallas y realizar actividades de sincronización y de seguridad. Los servidores proporcionan un servicio al cliente y devuelven los resultados.

En algunos casos existen procesos auxiliares que se encargan de recibir las solicitudes del cliente, verificar la protección, activar un proceso servidor para satisfacer el pedido, recibir su respuesta y enviarla al cliente. Además, deben manejar los interbloqueos, la recuperación ante fallas, y otros aspectos afines. Por las razones anteriores, la plataforma computacional asociada con los servidores es más poderosa que la de los clientes.

Para que los clientes y los servidores puedan comunicarse se requiere una infraestructura de comunicaciones, la cual proporciona los mecanismos básicos de direccionamiento y transporte. La mayoría de los sistemas cliente/servidor actuales, se basan en redes locales y por lo tanto utilizan protocolos no orientados a conexión, lo cual implica que las aplicaciones deben hacer las verificaciones. La red debe tener características adecuadas de desempeño, confiabilidad, transparencia y administración.

Como ejemplos de clientes pueden citarse interfaces de usuario para enviar comandos a un servidor, APIs para el desarrollo de aplicaciones distribuidas, herramientas en el cliente para hacer acceso a servidores remotos (por ejemplo, servidores de SQL) o aplicaciones que solicitan acceso a servidores para algunos servicios. Como ejemplos de servidores pueden citarse servidores de ventanas X Window, servidores de archivos como NFS, servidores para el manejo de bases de datos (como los servidores de SQL), servidores de diseño y manufactura asistidos por computadora, etc.

La red de terminales gráficas que se implementará en el laboratorio de electrónica, utilizará la arquitectura cliente-servidor, ya que contará con un servidor que proveerá los servicios o programas a los clientes que en este caso serán las terminales gráficas; las cuales usarán estos recursos para poder realizar alguna tarea de interés para el usuario. A su vez en un determinado momento, las terminales recibirán peticiones por parte del servidor para que cada una de ellas puedan desplegar en pantalla los resultados de los procesos.

2.4 Aspectos básicos en la instalación de Fedora Core Linux

Para garantizar el buen desempeño del servidor donde será implementado el LTSP y facilitar la instalación de Fedora Core, es necesario conocer detalladamente los principales aspectos que intervienen durante y posteriormente a la instalación de éste sistema operativo.

2.4.1 Tipos de instalación

Fedora Core proporciona cinco opciones de instalación, las cuales definen una serie de paquetes a instalar; es decir, dependiendo del uso que se le vaya a dar a la computadora y de los programas que se piensen utilizar, se instalará el software que más se adecue a las necesidades del usuario.

2.4.1.1 Escritorio personal

Es apropiado para estaciones de trabajo o portátiles, se debe seleccionar este tipo de instalación para crear un entorno gráfico y un sistema ideal para el uso del directorio principal o del escritorio; además es la más adecuada para principiantes.

2.4.1.2 Estación de trabajo

Esta opción instala un entorno gráfico con herramientas para el desarrollo del software y la administración del sistema.

2.4.1.3 Servidor

Este tipo de instalación es útil si se desea compartir archivos, impresoras, así como los diferentes servicios web. Se habilitarán también los servicios adicionales y se puede escoger si se desea o no instalar un entorno gráfico.

2.4.1.4 Personalizada

Con esta opción se tiene el control completo sobre el proceso de instalación, se podrá elegir el esquema de particionamiento, incluyendo la selección de los paquetes de software y las preferencias de la autenticación.

2.4.1.5 Actualización

Ésta es la solución más adecuada si ya se tiene instalada una versión de Fedora Core, y se quiere actualizar rápidamente a los últimos paquetes y versiones del kernel. A partir de las necesidades específicas de cada usuario, éste podrá elegir el tipo de instalación que más le conviene.

2.4.2 Métodos de instalación

Para instalar algún programa adicional en Linux, se deben efectuar ciertos pasos utilizando las herramientas apropiadas. Algunas distribuciones manejan sus propios métodos, pero los más conocidos y los que serán empleados en este trabajo serán los siguientes.

2.4.2.1 Código fuente (TGZ)

Las aplicaciones GNU son distribuidas en distintos formatos para realizar su instalación, entre ellos se encuentra su código fuente, no solo por el hecho de que se debe cumplir con la filosofía del software libre; si no también para permitir al usuario optimizar la aplicación a instalar para las características de su computadora.

Esta optimización se logra al compilar directamente el código fuente en la máquina donde se va a ejecutar el programa, ya que se toma en cuenta la arquitectura especifica de la misma, como el procesador, la memoria, etc. Para garantizar una compilación exitosa, es recomendable leer previamente los archivos de ayuda INSTALL y README, sin embargo por lo general los pasos para compilar el código fuente de una aplicación son los siguientes:

- Copiar el archivo .tgz, al directorio donde será descomprimido y compilado. Por ejemplo al directorio de root.

```
cp  archivo.tgz  /root
```

- Descomprimir el archivo: Para archivos con extensión .tgz o .tar.gz se utiliza el siguiente comando:

```
tar -zxvf  nombre_archivo
```

Para archivos con extensión .bz2 o .tar.bz2

```
tar -lxvf  nombre_archivo
```

- Se habrá creado un directorio con el nombre de la aplicación, se debe entrar a él con el comando:

```
cd nombre_directorio
```

- Se debe configurar el código fuente para determinar si se puede compilar y que lo hará sin errores. El archivo de configuración "configure" se utiliza para comprobar las dependencias de las bibliotecas necesarias; para garantizar que se cuente con ellas, es aconsejable que durante la instalación de Linux se seleccionen todas las herramientas de desarrollo, que incluyen generalmente el código fuente del software X, del Kernel, además de los entornos de escritorio GNOME y KDE.

  Se debe teclear lo siguiente:

```
./configure
```

  Esto creará los archivos "makefiles" necesarios para la compilación.

- Para iniciar la compilación solo se teclea lo siguiente:

```
make
```

Al final de la compilación se habrán generado el o los ejecutables optimizados para el sistema.

- Finalmente se instalan los ejecutables generados:

```
make install
```

2.4.2.2 RPM

El Administrador de paquetes Red Hat Package Manager (RPM) es un sistema de empaquetado de software que se utiliza en Fedora Core además de otras distribuciones Linux y UNIX. Los paquetes RPM se caracterizan por tener la extensión .rpm, el nombre del archivo incluye el nombre del paquete, su versión y la arquitectura para la cual fue optimizado.

Un RPM es un "paquete" o archivo que contiene (comprimido) un programa ejecutable, sus archivos de ayuda, los programas pre-instalados o dependencias que necesita para poder funcionar, algunos archivos de configuración y una serie de scripts para automatizar su instalación y desinstalación. RPM facilita la instalación y actualización del software, ya que es posible hacerlo por medio de comandos breves. Este sistema mantiene una base de datos de los paquetes y de sus archivos respectivos, por lo que se pueden realizar consultas y verificaciones de los paquetes ya instalados.

Los comandos básicos para el manejo de paquetes RPM, son los siguientes:

- Instalación:

```
rpm –ivh nombre_archivo.rpm
```

- Desinstalación:

```
rpm –e nombre_paquete_instalado
```

- Búsqueda de paquetes instalados:

```
rpm –qa | grep palabra_clave
```

2.4.3 Particionamiento

Las particiones garantizan que la capacidad de disco necesaria para determinados tipos de procesos no va a ser utilizada por otros. Esto ayuda a asegurar que los sistemas y procesos sigan ejecutándose, incluso si se saturan algunas particiones. Fedora Core proporciona 3 opciones de particionamiento:

- Particionamiento automático: Se configuran las particiones basándose en el tipo de instalación, espacio disponible en el disco duro, tamaño de la RAM y crea por default las siguientes particiones: Swap, Raiz (/), /boot. También se pueden personalizar las particiones restantes y adecuarlas si es necesario.

- Disk Druid: La herramienta del particionamiento manual del disco, permite configurar las particiones en un entorno interactivo. Se pueden configurar los tipos de sistemas de archivos, puntos de montaje, tamaño y otras opciones.

- Fdisk: Esta herramienta de particionamiento, es una utilidad de modo texto recomendada tan solo a usuarios avanzados que necesitan ejecutar tareas especializadas.

2.4.3.1 Particiones básicas en un sistema Linux

Todo sistema Linux debe tener un mínimo de 2 particiones, una para el sistema de archivos y por lo menos una partición de intercambio de memoria (swap). Se recomienda que al menos haya 2 particiones del sistema de archivos: una para los directorios que tiendan a crecer y la otra para los directorios que tengan que mantener espacio libre para que el sistema funcione. Para determinar el número y tamaño de las particiones en un sistema Linux, no hay alguna regla rigurosa; sin embargo, en algunos

textos se recomiendan configuraciones tanto para una estación de trabajo como para un servidor. [10]

- Particiones para una estación de trabajo

/	Partición root	500 MB – 1000 MB
/home	Archivos de usuario	1000 MB en adelante
swap	Memoria de intercambio	doble de RAM

- Particiones para un servidor

/boot	Partición boot (kernel/LILO)	2 MB – 100 MB
/	Partición root	80 MB – 100 MB
/usr	Partición de sistema	500 MB – 1000 MB
/usr/local	Partición para aplicaciones	500 MB en adelante
/opt	Partición para aplicaciones	500 MB en adelante
/home	Archivos de usuario	1000 MB en adelante
swap	Memoria de intercambio	Doble de RAM

Otra configuración propuesta es la siguiente:[11]

- Particiones para una estación de trabajo

/boot	Partición boot (kernel/LILO)	2 MB – 100 MB
/	Partición root	500 MB en adelante
swap	Memoria de intercambio	Doble de RAM

- Particiones para un servidor

/boot	Partición boot (kernel/LILO)	16 MB
/	Partición root	300 MB
/usr	Partición de sistema	300 MB – 700 MB
/home	Archivos de usuario	512 MB en adelante
/tmp	Datos temporales	200 MB en adelante
swap	Memoria de intercambio	Doble de RAM

En la práctica, si no se cuenta con suficiente espacio en el disco duro se puede reducir el tamaño de las particiones; tratando de que la disminución sea proporcional para todas. Incluso es posible omitir algunas, y Linux se encargará de gestionar el uso de las particiones existentes, pero esto incrementa el riesgo de que en algún momento si se corrompe alguna partición, se pierdan datos que originalmente deberían estar almacenados en otro lugar del disco.

Hasta este punto hemos conocido los diferentes componentes de red así como aspectos que son necesarios para la implementación del proyecto, ahora podremos instalar los programas que deben estar presentes en el servidor previamente a los paquetes del LTSP, así como su correcta configuración.

---

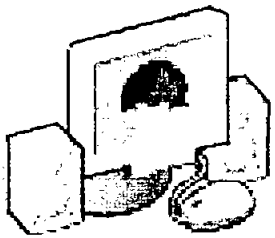
[10] KOPLER Michael, WESLEY Addison, Linux Installation, Configuration and Use, Ed. Great Britain, Great Britain, 2000, p.38

[11] PRITCHARD Kara J., Red Hat Certified Exam, Ed. Coriolis, USA, 2000, p.10

IMPLEMENTACIÓN

OBJETIVO ESPECÍFICO

Conocer las características y consideraciones generales para la instalación y configuración de los diferentes componentes tanto de hardware como de software, que intervienen en la implementación del LTSP.



3

INTRODUCCIÓN

En este capítulo se profundizará en los conocimientos técnicos específicos, así como las herramientas que serán necesarias para entender el funcionamiento, implementación y mantenimiento del LTSP.

Es importante definir lo que se necesita para comenzar con la implementación del proyecto, una parte fundamental es la elección de la máquina servidor, ésta debe contar con un buen procesador, suficiente memoria y un disco duro de capacidad aceptable. Principalmente se prevé todo esto para un mejor rendimiento, tomando en cuenta los servicios que va a ofrecer y a cuantas terminales podrá atender eficientemente.

Se enfatiza la forma de cómo redimensionar el disco duro para no perder la información que se tiene en Windows, debido a que se permitirá que el servidor preste servicios de estación de trabajo si así se requiere. Hay que recordar que una de las características importantes de Linux es que es capaz de leer y escribir en sistemas de archivos FAT, VFAT, FAT32, además de que encontramos infinidad de aplicaciones de libre distribución que se encuentran disponibles.

Se mostrará paso a paso cómo instalar Linux especificando las opciones que son requeridas para el ambiente LTSP, así como las que mejor se adapten a las necesidades propias del laboratorio.

Finalmente se instalará y configurará el LTSP, proporcionando las condiciones adecuadas para la siguiente fase del proyecto.

3.1 Requerimientos

El LTSP es un proyecto que requiere de ciertos componentes, los cuales deben cumplir con especificaciones que permitirán su adecuado funcionamiento y rendimiento al trabajar conjuntamente, para ello es necesario definir algunos requerimientos que permitan dar una referencia de lo que realmente se necesita para su implementación. A continuación se mencionarán los requerimientos recomendados de hardware y software.

3.1.1 Hardware

Una red de terminales gráficas debe contar con un servidor que cumpla con ciertas especificaciones, para que éste pueda atender de forma eficiente las peticiones que se le hagan. Las terminales demandan recursos del servidor para poder funcionar y por lo tanto su rendimiento depende de las capacidades que el servidor pueda ofrecer.

Existen otros factores que también determinan los requerimientos físicos del servidor, como son: el tipo de programas o servicios que se van a ejecutar, el número de usuarios que van a utilizar las terminales, la arquitectura de la red, entre otros.

Tomando en cuenta estos factores, se mencionarán los requerimientos de hardware recomendados para el servidor. A pesar de que con otras configuraciones se pueden obtener buenos resultados, incluso con un servidor de menor capacidad; es importante considerar que un servidor de mejores características implica un mayor rendimiento en las terminales.

Para un servidor con 5 terminales, los requerimientos recomendados son:[12]

- CPU Intel PIII, 1 GHz
- 512 MB RAM
- Disco duro 20 GB, IDE ATA 100/133
- (2) tarjetas de red Ethernet base 10/100
- 1 CD-ROM
- 1 Floppy

Para las terminales los requerimientos recomendados son:

- CPU 486 DX, 40MHz
- 16 MB RAM
- Tarjeta de red Ethernet base 10
- Tarjeta de video VGA
- 1 Floppy

[12] Pág. web del proyecto K12LTSP, Sección de Instalación http://k12ltsp.org/install.html

Para implementar un servidor LTSP se requiere tener al menos 256 MB de RAM, más 50 MB por cada terminal; algunas aplicaciones, por ejemplo las que manipulan imágenes, necesitarán más memoria por lo que se recomienda tener 1 GB.

Si la red tiene más de seis terminales, es recomendable contar con discos SCSI; estos discos se caracterizan por que son mucho más rápidos que los IDE, además de que permiten accesos múltiples a los mismos.

La velocidad del procesador también depende del uso que se le dará a los equipos; por ejemplo las aplicaciones de ofimática requieren de una velocidad relativamente pequeña (1 GHz). Al igual que con la memoria, las aplicaciones gráficas necesitan una mayor velocidad del procesador.

En cuanto a las tarjetas de red, las de 100 Mb funcionan bastante bien para una red que tenga hasta 30 terminales; además, es necesario contar con un Switch o un Hub, que permita implementar el modelo cliente-servidor. Se recomienda usar un Switch, ya que tiene control sobre colisiones y esto disminuye el tráfico en la red.

En el laboratorio de electrónica hay computadoras que satisfacen algunas de estas características; por lo que no será necesario adquirir algún equipo nuevo, además la idea de implementar este proyecto es precisamente el de seguir utilizando las computadoras con las que ya se cuenta, aunque no se descarta la posibilidad de que en un futuro se puedan mejorar las características del servidor.

Las características de este equipo son las siguientes:

Equipo 1

- CPU Intel 486, 66 MHz
- 16 MB RAM
- Disco duro 512 MB, IDE ATA 100
- Tarjeta de red 3Com 509 EtherLink III, bus ISA

Equipo 2

- CPU Intel 486, 66 MHz
- 16 MB RAM
- Disco duro 512 MB, IDE ATA 100
- Tarjeta de red 3Com 509 EtherLink III, bus ISA

Equipo 3

- CPU Intel PIII, 700 MHz
- 128 MB RAM
- Disco duro 20 GB IDE ATA 100
- Tarjeta de red MX98715, bus PCI

Equipo 4

- CPU Intel PIII, 700 MHz
- 384 MB RAM
- Disco duro 20 GB IDE ATA 100
- Tarjeta de red Sis 900, bus PCI
- Tarjeta de red RTL8139C, bus PCI

Se optó por utilizar el último equipo como servidor debido que cuenta con las mejores características.

3.1.2 Software

La computadora que será utilizada como servidor ya tiene instalado el sistema operativo Windows, sin embargo éste no reúne las características para implementar el LTSP; por esta razón es necesario instalar otro sistema operativo, para este caso Linux ya que cuenta con capacidades tanto de estación de trabajo como de servidor.

3.1.2.1 Redimensionar la partición

Debido a que en el laboratorio se trabaja con aplicaciones que solo pueden correr bajo Windows se optó por conservarlo, éste ocupa toda la capacidad del disco duro, es decir; solo existe una partición tipo FAT 32 sin embargo Linux requiere sus propias particiones.

Para liberar espacio en el disco duro donde estarán las particiones de Linux, se requiere modificar el tamaño de la partición de Windows, para lo cual se hará uso de la herramienta llamada Partition Magic que se ejecuta desde MS-DOS o Windows y permite crear, borrar o redimensionar las particiones de un disco de manera sencilla y confiable.

A continuación se muestra el proceso para liberar el espacio en disco para Linux:

Después de ejecutar el Partition Magic se muestra la partición de Windows que ocupa los 20 GB totales, pero se necesita redimensionarla para que solo sea de 10 GB, el espacio restante será asignado a las particiones de Linux.

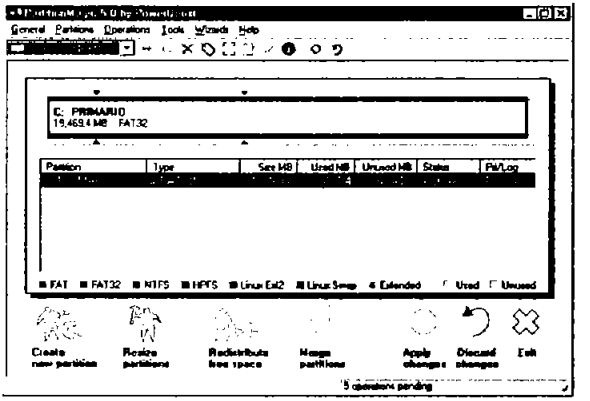


Figura 3-1 Se muestra la partición de Windows que ocupa 20 GB

Posteriormente se elige resize/move del menú operations, que muestra una ventana donde se pueden modificar las propiedades de la partición.

El tamaño de la partición se puede cambiar de 2 maneras, la primera posicionándose al final de la barra y arrastrando el puntero del ratón hasta fijar el nuevo tamaño y la segunda introduciendo el valor correspondiente en MB en el campo New Size, por último se da click en el botón OK para aceptar los cambios.

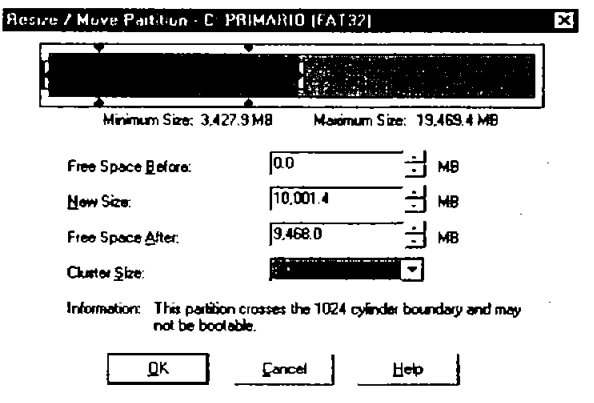


Figura 3-2 Proceso de liberación de espacio en disco para Linux

Ahora se puede observar el nuevo tamaño de la partición de Windows y el espacio libre para Linux.

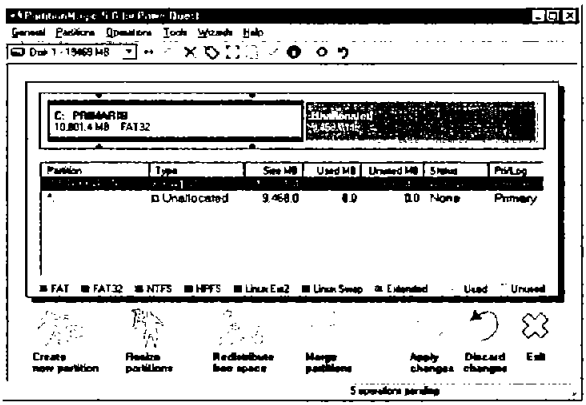


Figura 3-3 Nuevo tamaño de la partición de Windows y el espacio para Linux

Finalmente se da click en Apply Changes para efectuar los cambios, el programa tendrá que cerrar Windows para comenzar con el proceso de redimensionamiento. Una vez que haya finalizado este proceso, se podrá comenzar con la instalación de Linux. Si no se desea conservar la instalación de Windows, se puede formatear el disco duro, para que Linux ocupe todo el espacio disponible.

3.1.2.2 Instalación del sistema operativo

En esta sección se describen las opciones y configuraciones recomendadas durante la instalación de Fedora Core. En este caso el método de instalación será vía CD-ROM, por lo que es necesario configurar el BIOS de la computadora para que permita bootear desde este dispositivo.

Paso 1: Se introduce el CD 1 y automáticamente se inicia el programa de instalación de Fedora Core llamado Anaconda, en caso de que el programa no pueda ser iniciado se tiene que crear un disco de arranque, éste permitirá cargar una imagen del sistema operativo con la información necesaria para poder realizar la instalación por medio del CD-ROM.

El disco de arranque se crea de la siguiente manera:

Desde MS-DOS, se utiliza el comando rawrite, el cual viene incluido en el primer disco de instalación de Fedora Core, en la carpeta DOSUTILS.

Se escribe el comando:

```
D:\DOSUTILS\rawrite
```

Se pedirá la ruta de la imagen de arranque:

```
D:\IMAGES\boot.img
```

Y después la ruta donde se escribirá la imagen de arranque, en este caso es el disquete:

```
A:\
```

Se tiene que reiniciar la computadora para que ésta arranque por medio del disco generado.

Paso 2: El programa de instalación muestra una pantalla de inicio, donde no se pide información alguna. La sección de ayuda aparece por defecto, es recomendable leer la información mostrada.

Para continuar con la instalación, se debe hacer click en el botón "Siguiente".

Paso 3: Seleccionar el idioma español, éste es el lenguaje que será utilizado para la instalación y para el sistema. Hacer click en "Siguiente" para continuar.

Paso 4: Seleccionar el tipo de teclado apropiado para el sistema, en este caso se elige "español". Hacer click en "Siguiente" para continuar.

Paso 5: Se debe seleccionar el tipo de ratón, en la mayoría de los casos será detectado automáticamente. Si el ratón es de 2 botones, se puede habilitar la opción para emular 3 botones. Hacer click en "Siguiente" para continuar con el proceso.

Paso 6: Debido a la flexibilidad que se requiere para la instalación de un servidor de terminales gráficas, se debe seleccionar el tipo de instalación más apropiado; en este caso será la instalación "personalizada". Para continuar, se debe hacer click en "Siguiente".

Paso 7: En este punto se tiene que elegir una de las opciones de particionamiento del disco duro. Seleccionar "Disk Druid" y hacer click en "Siguiente".

Paso 8: Se debe tener cuidado al crear las particiones ya que de esto depende el buen funcionamiento del sistema.

Disk Druid muestra gráficamente las particiones del disco duro, en este caso la partición de Windows y el espacio libre destinado para Linux.

La primer partición que hay que crear es la swap, para esto se debe seleccionar con el ratón el espacio libre del disco duro, y hacer click en el botón "nuevo" lo cual abre un cuadro de diálogo donde se puede configurar las características de la nueva partición.

Se debe seleccionar en el campo "Tipo de sistema de archivos" la opción "Swap", posteriormente se especifica su tamaño, el cual será de 768 MB. Finalmente se habilita la opción "tamaño fijo". Hacer click en Aceptar para cerrar el cuadro de diálogo.

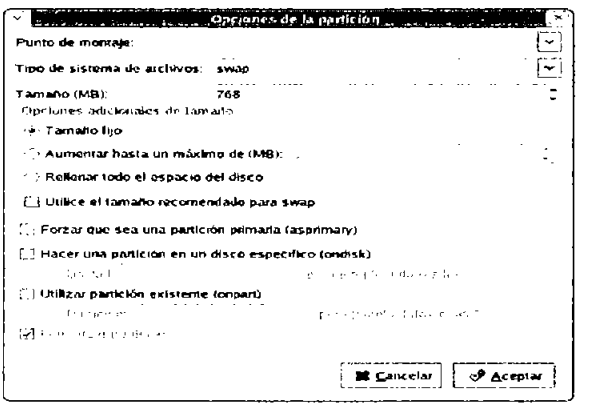


Figura 3-4 Proceso de creación de la partición swap

La segunda partición que debe ser creada es la /boot, que se utiliza para almacenar los archivos de arranque del sistema.

Con ayuda del ratón se debe seleccionar el espacio libre del disco duro, y hacer click en el botón "nuevo" lo cual abre el cuadro de diálogo para configurar la nueva partición.

En la sección punto de montaje se debe elegir "/boot" y como sistema de archivos el tipo "ext3". Esta partición tendrá un tamaño de 100 MB, por lo que se debe habilitar la opción "tamaño fijo".

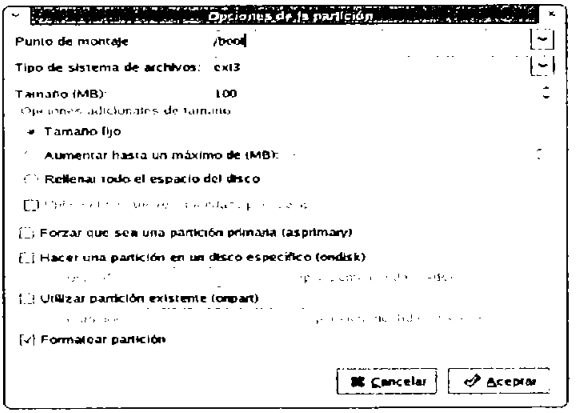


Figura 3-5 Proceso de creación de la partición boot

La última partición que se tiene que crear es la raíz (/), para esto debe seleccionarse el espacio libre del disco duro, posteriormente se hace click en el botón "nuevo" para abrir el cuadro de diálogo correspondiente.

En la sección "punto de montaje" se elige "/" y como sistema de archivos el tipo "ext3", en la sección "opciones de tamaño" se debe seleccionar "completar hasta el tamaño máximo permitido"; lo cual permite seleccionar todo el espacio libre del disco duro.

Hacer click en Aceptar para cerrar el cuadro de diálogo.

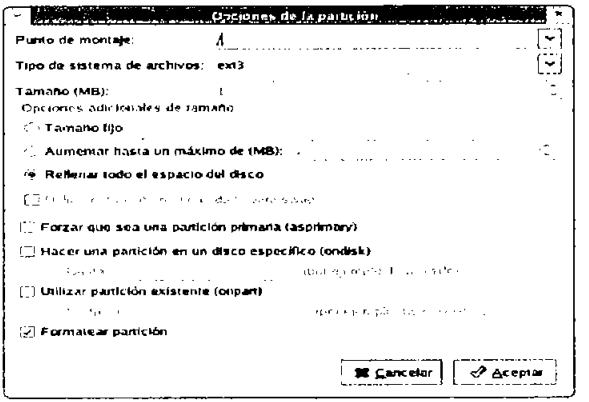


Figura 3-6 Proceso de creación de la partición raíz

Para finalizar el proceso de creación de las particiones de debe hacer click en "siguiente".

Paso 9: El programa Anaconda, instala por defecto el gestor de arranque GRUB, el cual permitirá elegir con que sistema operativo iniciar (Windows o Fedora Core). Hacer click en "Siguiente" para continuar.

Paso 10: Fedora Core permite proteger el sistema mediante un firewall, para instalar un nivel aceptable de seguridad, se debe elegir la opción "nivel intermedio". Hacer click en "Siguiente" para continuar.

Paso 11: Es necesario instalar un idioma para usarlo por defecto, éste será utilizado por el sistema una vez que la instalación se haya completado. En este caso se debe seleccionar "Español (México)". Hacer click en "Siguiente" para continuar.

Paso 12: El programa de instalación permite seleccionar un huso horario, el cual será "América / Ciudad de México". Hacer click en "Siguiente".

Paso 13: En este paso se tiene que proporcionar la cuenta root, la cual es usada para instalar paquetes, actualizar RPM's y realizar la mayoría de las tareas de mantenimiento del sistema. Una vez que se haya ingresado y confirmado la contraseña, se debe hacer click en "Siguiente".

Paso 14: En esta parte es donde se eligen los paquetes que serán instalados en el sistema.

A continuación se muestra una lista con los grupos de paquetes a instalar.

- Escritorios
  - Sistema X Window
  - GNOME
  - KDE

- Aplicaciones
  - Editores
  - Ingeniería y científico
  - Internet gráfica
  - Oficina/productividad
  - Sonido y video
  - Gráficos

- Servidores
  - Servidores de red (elegir DHCP)

- Desarrollo
  - Herramientas de desarrollo
  - Desarrollo del kernel

- Desarrollo del software X
- Desarrollo de software de GNOME
- Desarrollo de software para KDE

➢ Sistema
- Herramientas de administración
- Soporte para la impresión

Paso 15: El programa de instalación formateará las particiones e iniciará la instalación de los paquetes necesarios. Durante esta etapa, se le solicitará el ingreso de los CD's 2 y 3.

Paso 16: Una vez finalizado el proceso de instalación de los paquetes, se puede crear un disco de arranque; este disco se utiliza en caso de que el gestor de arranque GRUP no funcione. Hacer click en continuar.

Paso 17: Una vez que se hayan completado todos los pasos anteriores, el programa finalizará la instalación.

3.2 Instalación y configuración de los paquetes en el servidor.

Hasta ahora se conoce de manera general cual es la forma en que trabaja el LTSP, sin embargo es necesario comprender a fondo todo lo que implica la implementación de este proyecto; para ello se describirá la función y la instalación de cada uno de los programas que deben estar presentes en el servidor previamente a los paquetes del LTSP. Posteriormente se mostrará la forma en que deben ser configurados para que funcionen en conjunto con el LTSP.

Se da por hecho que el servidor ya debe estar corriendo el sistema operativo Fedora Core, y que se cuenta con los permisos de root.

3.2.1 DHCP

Dynamic Host Configuration Protocol (Protocolo de Configuración Dinámica de Servidor) es un protocolo de red que se utiliza para asignar automáticamente información TCP/IP a equipos cliente. Cada uno de ellos se conecta a un servidor DHCP centralizado que devuelve la configuración de red del cliente, incluida la dirección IP, el gateway y los servidores DNS.

DHCP es útil para proporcionar de un modo rápido la configuración de red de una máquina cliente, al configurar el sistema cliente el administrador puede seleccionar

el protocolo DHCP y no especificar una dirección IP, una máscara de red, un gateway o servidor DNS fijos; el cliente recupera esta información desde el servidor.

DHCP también es útil si un administrador desea cambiar la dirección de IP de muchos sistemas; en lugar de volver a configurarlos todos, puede modificar un archivo de configuración en el servidor para establecer la nueva dirección IP.

Si los servidores DNS de una organización cambian, los cambios también se aplicarán en el servidor DHCP, no en todos los clientes. Una vez que se reinicie la red, se aplicarán los cambios.

Además, si una computadora portátil o cualquier tipo de equipo móvil se configura para hacer peticiones DHCP, podrá desplazarse entre distintas oficinas sin tener que volver a configurarse, ya que cada oficina dispondrá de un servidor DHCP que permitirá su conexión a la red.

3.2.1.1 Instalación de los paquetes RPM

Para la instalación del servicio DHCP, se debe descargar la última versión de los paquetes y ejecutar los siguientes comandos:

```
rpm -ivh dhcp-devel-3.0pl1-9.i386.rpm
rpm -ivh dhcp-3.0pl1-9.i386.rpm
```

3.2.2 TFTP

Trivial File Transfer Protocol (Protocolo Simple de Transferencia de Archivos) es un protocolo que utiliza el UDP (Protocolo de Datagramas de Usuario) para transferir archivos entre distintas máquinas conectadas a través de una red.

Se implementa sobre un servicio de comunicaciones no fiable y no orientado a conexión, por lo que no requiere el nombre de un usuario ni su contraseña, lo que da lugar a que carezca de seguridad. TFTP ejecuta las mismas tareas que el FTP (Protocolo de Transferencia de Archivos), pero usa un protocolo de transporte diferente.

3.2.2.1 Instalación de los paquetes RPM

Para la instalación del servicio TFTP, se debe descargar la última versión de los paquetes y ejecutar los siguientes comandos:

```
rpm -ivh tftp-0.29-3.i386.rpm
rpm -ivh tftp-server-0.29-3.i386.rpm
```

3.2.3 NFS

Network File System (Sistema de Archivos de Red) es un método para compartir archivos entre máquinas de una red, de tal forma que parece que se está trabajando en un disco duro local. Esto permite centralizar archivos en una localización, mientras se permite su acceso continuo a los usuarios autorizados.

Fedora Core puede trabajar como servidor o como cliente de NFS (o ambos), lo que implica que puede exportar sistemas de archivos a otras máquinas, así como montar los sistemas de archivos que otras computadoras exportan.

NFS resulta útil para compartir directorios de archivos entre múltiples usuarios de la misma red. Por ejemplo, un grupo de usuarios que trabajan en un mismo proyecto pueden tener acceso a los archivos del mismo usando una porción compartida del sistema de archivos NFS, que se ha montado en un directorio determinado, como pudiera ser /myproject. Para acceder a los archivos compartidos; el usuario accede al directorio /myproject de su máquina local.

Linux usa una combinación de soporte a nivel de kernel y demonios en continua ejecución para proporcionar la disponibilidad de los archivos vía NFS, y el servicio debe estar activo en el kernel de Linux para que funcione. NFS usa Remote Procedure Calls (Llamadas de Procedimientos Remotos) para enrutar peticiones entre clientes y servidores, implicando que el servicio portmap deba estar disponible y activo en los niveles de ejecución adecuados para que la comunicación funcione.

Con NFS, la autentificación solo se produce cuando el cliente intenta montar un sistema de archivos remoto. Para limitar el acceso, el servidor NFS utiliza los archivos /etc/hosts.allow y /etc/hosts.deny para determinar si a un cliente en particular le debe ser explícitamente permitido o denegado su acceso.

Después de revisar los archivos /etc/hosts.allow y /etc/hosts.deny, el servidor NFS recurre a su archivo de configuración, /etc/exports, para determinar si el cliente tiene suficientes privilegios para montar alguno de los sistemas de archivos exportados. Los privilegios de montaje NFS son permitidos específicamente a clientes, no a usuarios. Si se permite a una máquina acceder a un sistema de archivos exportado, los usuarios de esa máquina podrán acceder a esos datos.

3.2.3.1 Instalación de los paquetes RPM

Para la instalación del servicio NFS, se debe descargar la última versión de los paquetes y ejecutar los siguientes comandos:

```
rpm -ivh nfs-utils-1.0.1-2.i386.rpm
rpm -ivh portmap-4.0-46.i386.rpm
```

3.2.4 LTSP

Existen una serie de paquetes para instalar el entorno LTSP, hay 3 grupos principales los cuales son: Core, Linux Kernel y X Windows. Se dividieron los paquetes en grupos, para mejorar su funcionalidad y flexibilidad.

3.2.4.1 Paquete Core

Este paquete proporciona los scripts de instalación del LTSP, los cuales permiten crear la estructura de directorios que será compartida a cada una de las terminales y también los archivos plantilla para la configuración de los servicios requeridos.

Se deberá conseguir la última versión del paquete y ejecutar el siguiente comando:

```
rpm –ivh ltsp_core-3.0.7-4.i386.rpm
```

3.2.4.2 Paquete Ltsp_Kernel

Este paquete contiene un kernel con un ramdisk (disco virtual). Dentro del ramdisk se incluye soporte para todos los módulos de las tarjetas de red y un espacio de usuario con un cliente dhcp llamado dhclient.

Se deberá conseguir la última versión del paquete y ejecutar el siguiente comando:

```
rpm –ivh ltsp_kernel-3.0-5.i386.rpm
```

3.2.4.3 Paquete X Windows

Este paquete es necesario si se desea configurar una terminal gráfica (Interfaz gráfica GUI), el cual proporciona todos los drivers pertenecientes al XFree86; es recomendable que también se instale el paquete de fuentes X Fonts. Este paquete viene separado del X Core debido a que en la mayoría de los casos se utiliza un servidor de fuentes X (Xfs).

Se deberá conseguir la última versión de los paquetes y ejecutar los siguientes comandos:

```
rpm –ivh ltsp_x_core-3.0.4-2.i386.rpm
rpm –ivh ltsp_x_fonts-3.0.4-2.i386.rpm
```

3.2.4.4 Scripts de configuración LTSP

Después de instalar los paquetes principales se requiere ejecutar algunos scripts que son necesarios para configurar los archivos de sistema del servidor. Cada uno de estos scripts es responsable de modificar o en su caso crear archivos de ejemplo de los servicios NFS, TFTP, DHCP, etc.

Se accede al directorio install_scripts con el siguiente comando:

```
cd /opt/ltsp/install_scripts
```

Posteriormente se ejecuta el siguiente script para crear el directorio templates.

```
sh ./install.sh
```

Este script muestra la siguiente salida:

```
About to install LTSP, using the following settings:

 LTSP_DIR    = /opt/ltsp
 SWAP_DIR    = /var/opt/ltsp/swapfiles
 TFTP_DIR    = /tftpboot
 IP_NETWORK  = 192.168.0.0
 IP_SERVER   = 192.168.0.254
 IP_NETMASK  = 255.255.255.0
 IP_BROADCAST = 192.168.0.255

If you want to install LTSP using the above settings,
enter 'Y' and the installation will proceed.  Any other
response will abort the installation, and you can modify
the CONFIG file and restart the installation.

Continue with installation (y/n)?

find: i386: No existe el archivo o el directorio
./install.sh: line 274: /opt/ltsp/i386/etc/version: No existe el archivo o el directorio

Take a look in /tmp/ltsp.install.log for a complete log of the installation

You now need to change to the /opt/ltsp/templates directory and
run the ltsp_initialize script to complete the installation

./install.sh: line 397: /opt/ltsp/i386/etc/lts.orig: No existe el archivo o el directorio
```

Tabla 3-1 Creación del directorio templates

Aunque se muestran ciertos errores, estos pueden ignorarse; lo importante en este punto es que se haya creado el directorio templates.

Una vez creado el directorio se tendrá que entrar a él.

```
cd ../templates
```

Estando en el directorio templates se ejecuta lo siguiente:

```
./ltsp_initialize
```

Este comando preguntará qué servicios se quieren configurar, para ello tendrá que crear o modificar ciertos archivos. La salida que muestra este comando es la siguiente:

```
Linux Terminal Server Project (http://www.LTSP.org)

The following files will be created/modified:

/etc/X11/xdm/Xaccess              The config file to allow remote xdm log  [Y]
/etc/X11/xdm/Xsetup_workstation   Sets the logo of your login window       [Y]
/etc/dhcpd.conf.example           Example config file for dhcp             [Y]
/etc/exports                      The config file for nfs                  [Y]
/etc/X11/gdm/gdm.conf             The config file for gdm                  [Y]
/etc/X11/gdm/Init/Default         The gdm startup script                   [Y]
/etc/hosts.allow                  Configuration file for tcp wrappers      [Y]
/etc/inittab                      Config file for init                     [Y]
/etc/kde/kdm/kdmrc                The config file for kdm                  [Y]
/etc/X11/xdm/ltsp.gif             The background logo for your login       [Y]
/etc/rc.d/rc5.d/S60nfs            Startup links for nfs                    [Y]
/etc/rc.d/rc5.d/S13portmap        Startup links for portmapper             [Y]
/etc/sysconfig/syslog             Startup info for syslogd                 [Y]
/etc/xinetd.d/tftp                Enable the tftp daemon                   [Y]
/etc/X11/xdm/xdm-config           The main config file for xdm/kdm         [Y]


Ready to apply the changes? ( R-Review, A-Apply, C-Cancel )
```

Tabla 3-2 Ejecución de scrip ltsp.initialize en el directorio templates

Debido a que en ocasiones algunos de estos archivos son configurados manualmente por el administrador por cuestiones de seguridad, es recomendable verificar las modificaciones que realiza este script. En caso de estar convencidos de que no se va a comprometer la seguridad del sistema, se pueden aplicar los cambios; lo que generará o en su caso modificará los archivos mencionados.

3.2.5 Configuración de los archivos de sistema

Los scripts del LTSP, crearán entradas (líneas de código) en los archivos de configuración de los diferentes servicios necesarios; estas entradas son configuraciones por default de lo que se necesita para que el servidor LTSP funcione adecuadamente. Algunos parámetros podrán modificarse o adaptarse dependiendo de la configuración de la red, como por ejemplo las direcciones IP; sin embargo es recomendable conservar los valores iniciales. En algunos casos se crearán líneas comentadas, las cuales servirán de ejemplo en caso de requerir configuraciones especiales, esto sucede a menudo debido a la gran diversidad del hardware disponible. A continuación se muestran los archivos de sistema más importantes para la implementación del LTSP.

3.2.5.1 dhcpd.conf

Este archivo de configuración consta de un conjunto de sentencias, las cuales se clasifican en parámetros y declaraciones. Los parámetros expresan como hacer algo, si se hace algo o no, así como los atributos que se le asignan al cliente. Las declaraciones, en cambio, se emplean para describir la topología de una red, describir a un conjunto de clientes o para aplicar determinados parámetros a un grupo de declaraciones.

Las declaraciones tienen la forma:

```
<nombre de la declaración> [atributos] {
        [parámetros]
        [declaraciones]
}
```

y los parámetros:

```
[option] <nombre del parámetro> [valores];
```

Los parámetros que comienzan con la palabra reservada "option" describen aquellos datos que brinda el servidor al cliente como parte del protocolo, y los que no, describen las características del servidor de DHCP.

A continuación se describen las sentencias declarativas:

- shared-network: Permite agrupar un conjunto de subredes que compartan la misma red física. El único atributo de esta sentencia es un nombre que sólo se utiliza para las trazas del servicio.

Sintaxis:

```
shared-network <nombre> {
        [parámetros]
        [declaraciones]
}
```

- subnet: Permite agrupar las características globales que van a tener los clientes de una misma subred.

Sintaxis:

```
subnet <dirección de red>netmask<máscara de red> {
        [parámetros]
        [declaraciones]
}
```

- range: Permite definir un rango de direcciones IP a otorgar a clientes pertenecientes a una subred. Toda declaración tipo subnet debe tener asociada una declaración range en la cual se especifique las direcciones IP mínima y máxima. Si se especifica el atributo dynamic-bootp se indica que estas direcciones se pueden asignar también a clientes BOOTP. Cuando se especifica una sola dirección IP se omite la dirección máxima.

Sintaxis:

```
range [dynamic-bootp] <dirección IP mínima> [dirección IP máxima]
```

- host: Permite describir aquellos hosts que tengan una dirección fija. Todos los clientes que usan BOOTP deben tener asociada una sentencia host. Un cliente se corresponde con una declaración host si la opción dhcp-client-identifier indicada en la declaración posee el valor del identificador que brinda el cliente a través del protocolo. De no ser así entonces se emplearía la dirección MAC del cliente especificada a través del atributo hardware.

Sintaxis:

```
host <hostname> {
        [parámetros]
        [declaraciones]
}
```

- group: Permite agrupar a otras declaraciones para aplicarles varios parámetros comunes. Puede ser utilizada para agrupar hosts, subredes, redes compartidas y otros grupos.

Sintaxis:

```
group {
        [parámetros]
        [declaraciones]
}
```

Los principales parámetros son:

- lease-file-name <filename>: Indica el nombre del archivo donde se almacenan los contratos. Este parámetro tiene alcance global por lo que se debe especificar fuera de todos los ámbitos (declaraciones) para que tenga efecto real. Por defecto es /var/lib/dhcp/dhcpd.leases

- default-lease-time <time>: Expresa en segundos la duración de una conexión.

- max-lease-time <time>: Expresa en segundos la duración máxima de una conexión.

- min-lease-time <time>: Expresa en segundos la duración mínima de una conexión.

- min-seconds <seconds> : Indica el número de segundos que debe esperar el servidor DHCP para responder a la petición de los clientes. Se utiliza cuando se tiene un segundo servidor y se desea que este responda después que el otro haya atendido la petición del cliente.

- hardware <type> <address>: Indica la dirección física (MAC) de un cliente particular (declaraciones tipo host). El atributo type expresa el tipo de arquitectura de la interfaz de red, actualmente puede ser: ethernet o token-ring.

- server-name <servername>: Indica el nombre que se ofrecerá a los clientes como identificador del servidor que emplean.

- fixed-address <address> [, <address>]: Expresa las direcciones IP que son fijas para los clientes descritos a través de las declaraciones de tipo host. Pueden utilizarse nombres de dominio en lugar de números IP.

- dynamic-bootp-lease-cutoff <date>: Indica la fecha en que expiran los contratos de todos los clientes BOOTP.

- get-lease-hostnames <flag>: Indica si el servidor resolverá o no las direcciones IP de los clientes a nombres de dominio y usará estos nombres como la opción host-name del protocolo.

- use-host-decl-names <flag>: Indica si se asume que el nombre provisto en cada una de las declaraciones tipo host dentro del mismo ámbito, es el nombre del cliente correspondiente (opción host-name del protocolo).

- authoritative;
- non authoritative;

Indican si el servidor está autorizado o no para realizar sus funciones. Por defecto un servidor DHCP asume que la información que brinda a una subred determinada no es correcta ni tampoco está autorizado para brindar el servicio.

Esto permite que si un usuario inexperto instala un servidor de DHCP en la red este no sea escuchado por los clientes como lo es un servidor legítimo que se le indique explícitamente que está autorizado.

El administrador de red que configure adecuadamente su servidor debe colocar este parámetro al comienzo del archivo, aunque puede ser conveniente en algunas ocasiones declarar al servidor autorizado de acuerdo a los segmentos de red definidos y no de forma global.

- always-broadcast <flag>: Se emplea para algunos clientes de DHCP/BOOTP que no pueden recibir las respuestas del servidor si no son en forma de broadcast. Se debe tratar de colocar este parámetro a "on" sólo para los clientes que realmente lo necesiten pues provoca demasiado tráfico en la red.

- ddns-update <flag>: Indica si se realizan o no actualizaciones dinámicas al DNS siempre que se establezca un contrato. Por defecto este parámetro tiene valor "on".

- allow <request>;
- deny <request>;
- ignore <request>;

Se emplean para controlar la respuesta del servidor DHCP ante distintos tipos de peticiones, algunas de las posibles solicitudes (atributo request) son:

  - unknown-clients: se emplea para indicar al servidor si acepta o no las solicitudes de los clientes desconocidos. Un cliente desconocido es aquel que no tiene asociado una declaración tipo host. Por defecto las solicitudes de estos clientes se aceptan.

- bootp: se utiliza para señalar si se aceptarán o no los pedidos de los clientes BOOTP. Por defecto se aceptan.

- booting: se emplea en las declaraciones del tipo host para indicar si se aceptará o negará la solicitud del host correspondiente. Por defecto se aceptan para todos los hosts.

- declines: se utiliza para indicar si el servidor acepta o no los mensajes del tipo DHCPDECLINE de los clientes. Cuando un servidor recibe este tipo de mensajes asume que la dirección que ofrece no es válida pues al parecer alguien no autorizado la está utilizando y entonces la declara como abandonada.

  Desafortunadamente un cliente "malicioso" o con una implementación incorrecta puede agotar todo el spool de direcciones a otorgar que posee el servidor y antes de que este decida emplear las direcciones abandonadas ya se habrán provocado algunos trastornos en el servicio.

Entre los parámetros que se le pueden otorgar a un cliente a través del protocolo y que van precedidos por la palabra "option", se encuentran:

- option domain-name <domain name>: Indica el nombre del dominio que empleará el cliente.

- option domain-name-servers <ip address> [, <ip address> ...]: Indica los servidores de nombres de dominio a emplear por el cliente.

- option host-name <hostname>: Indica el nombre que empleará el host cliente.

- option subnet-mask <ip address>: Indica la máscara de red que se le asignará al cliente.

- option routers <ip address> [, <ip address> ...]: Indica las direcciones IP de los routers (gateway) que empleará el cliente.

- option broadcast-address <ip address>: Indica la dirección de broadcast que utilizará el cliente.

- option dhcp-client-identifier <string>: Indica el identificador que puede emplear el cliente como alternativa a su dirección MAC.

Un ejemplo del archivo dhcpd.conf es creado como parte de la instalación del LTSP. Este archivo es llamado /etc/dhcpd.conf.example y puede ser copiado o renombrado a /etc/dhcpd.conf.

El archivo tiene la siguiente estructura:

```
# Sample configuration file for dhcpd
#
# Make changes to this file and copy it to /etc/dhcpd.conf
#
ddns-update-style       none;

default-lease-time          21600;
max-lease-time              21600;

option subnet-mask          255.255.255.0;
option broadcast-address    192.168.0.255;
option routers              192.168.0.254;
option domain-name-servers  192.168.0.254;
option domain-name          "yourdomain.com";
option root-path            "192.168.0.254:/opt/ltsp/i386";

option option-128 code 128 = string;
option option-129 code 129 = text;

shared-network WORKSTATIONS {
  subnet 192.168.0.0 netmask 255.255.255.0 {
  }
}

group   {
  use-host-decl-names     on;
  option log-servers      192.168.0.254;

  host ws001 {
    hardware ethernet   00:E0:06:E8:00:84;
    fixed-address       192.168.0.1;
    filename            "/lts/vmlinuz-2.4.19-ltsp-1";
##      option option-128   e4:45:74:68:00:00;  #This is NOT a MAC address
##      option option-129   "NIC=ne IO=0x300";
  }
  host ws002 {
    hardware ethernet   00:D0:09:30:6A:1C;
    fixed-address       192.168.0.2;
    filename            "/lts/vmlinuz-2.4.9-ltsp-6";
  }
}
```

Tabla 3-3 Ejemplo de configuración del archivo dhcpd

Una vez renombrado el archivo y hechas las modificaciones pertinentes el dhcpd puede ser iniciado con el siguiente comando:

```
service dhcpd start
```

3.2.5.1.1 Terminal 1

Para este equipo, la configuración del archivo dhcpd.conf debe ser:

```
host ws001 {
        hardware ethernet    00:60:8C:CA:BC:FF;
        fixed-address        192.168.0.1;
        filename             "/lts/vmlinuz-2.4.19-ltsp-1";
        option option-128    e4:45:74:68:00:00;
        option option-129    "NIC=3c509";
}
```

3.2.5.1.2 Terminal 2

Para este equipo, la configuración del archivo dhcpd.conf debe ser:

```
host ws002 {
        hardware ethernet    00:A0:24:24:F6:0F;
        fixed-address        192.168.0.2;
        filename             "/lts/vmlinuz-2.4.19-ltsp-1";
        option option-128    e4:45:74:68:00:00;
        option option-129    "NIC=3c509";
}
```

3.2.5.1.3 Terminal 3

Para este equipo, la configuración del archivo dhcpd.conf debe ser:

```
host ws003 {
        hardware ethernet    00:80:AD:3B:8F:E4;
        fixed-address        192.168.0.3;
        filename             "/lts/vmlinuz-2.4.19-ltsp-1";
}
```

3.2.5.2 tftp

El archivo tiene la siguiente estructura:

```
# default: off
# description: The tftp server serves files using the trivial file transfer\
#       protocol.  The tftp protocol is often used to boot diskless \
#       workstations, download configuration files to network-aware printers, \
#       and to start the installation process for some operating systems.
service tftp
{
        socket_type             = dgram
        protocol                = udp
        wait                    = yes
        user                    = root
        server                  = /usr/sbin/in.tftpd
        server_args             = -s /tftpboot
        disable                 = yes
        per_source              = 11
        cps                     = 100 2
        flags                   = IPv4
}
```

Tabla 3-4 Ejemplo de configuración del archivo tftp

En Fedora Core, el servicio tftp es iniciado por xinetd; el archivo de configuración tftp contiene la información necesaria para que corra tftpd. Por lo que el servicio podrá ser iniciado con el siguiente comando:

```
service xinetd start    ó
service xinetd restart
```

3.2.5.3 hosts

Por lo general las computadoras se comunican con direcciones IP, pero para las personas es más sencillo recordar nombres que números, es entonces cuando el DNS o el archivo /etc/hosts entra escena. Este mapeo de direcciones IP a nombres de host no es requerido generalmente, pero en un ambiente LTSP si lo es. Esto es así porque sin este mapeo, el NFS dará errores de permisos cuando la terminal intente montar el sistema de archivos raíz. Aparte de los problemas de NFS, si la terminal no se encuentra en el archivo /etc/hosts, también se podrían tener problemas con los Display Managers GDM y KDM.

El principal propósito de este archivo es resolver los nombres de hosts a partir de su dirección IP. Se puede usar solamente para resolver nombres de hosts en pequeñas redes sin servidor DNS.

Sin tener en cuenta el tipo de red que la computadora utilice, este archivo contiene una línea que especifica la dirección IP del dispositivo loopback (127.0.0.1) como por ejemplo localhost.localdomain; éste dispositivo hace referencia a la misma computadora.

La sintaxis de este archivo es muy simple:

```
<dirección IP> minombre.midominio.org alias
```

Se deben utilizar direcciones IP que pertenezcan al rango de direcciones IP privadas. Los rangos válidos son:

Clases de redes

A 10.0.0.0
B Entre 172.16.0.0 y 172.31.0.0
C Entre 192.168.0.0 y 192.168.255.0

El archivo tiene la siguiente estructura:

```
# Do not remove the following line, or various programs
# that require network functionality will fail.
127.0.0.1       localhost.localdomain       localhost
192.168.0.254   localhost.localdomain       localhost
```

Tabla 3-5 Ejemplo de configuración del archivo hosts

3.2.5.3.1 Terminal 1

Para este equipo, la configuración del archivo hosts debe ser:

```
192.168.0.1     ws001     ws001.localdomain     ws001
```

3.2.5.3.2 Terminal 2

Para este equipo, la configuración del archivo hosts debe ser:

```
192.168.0.2     ws002     ws002.localdomain     ws002
```

3.2.5.3.3 Terminal 3

Para este equipo, la configuración del archivo hosts debe ser:

```
192.168.0.3     ws003     ws003.localdomain     ws003
```

3.2.5.4 Hosts.allow, hosts.deny

En un servidor Linux el acceso a los servicios que usan wrappers TCP y que se basan en el nombre de la máquina, lo controlan dos archivos: /etc/hosts.allow y /etc/hosts.deny. Estos archivos usan un formato simple para controlar el acceso a los servicios.

Siempre y cuando no se haya especificado alguna regla en hosts.allow y hosts.deny, la regla por defecto consiste en permitir a cualquiera el acceso a los servicios.

El orden es importante ya que las reglas que se encuentran en hosts.allow tienen precedencia con respecto a las especificadas en hosts.deny. Aunque una regla deniegue el acceso a un determinado servicio en hosts.deny, los clientes que tengan el permiso para acceder al servicio en hosts.allow lo mantienen.

Las reglas de cada uno de los archivos están colocadas por orden de importancia por lo que hay que respetar dicho orden. Todos los cambios que se hagan en estos archivos afectan inmediatamente a los servicios a los que se refieran; no se requiere volver a arrancar los servicios. Todas las reglas del control de acceso que se encuentran en las líneas hosts.allow y hosts.deny, y los espacios en blanco o las líneas que empiezan con (#) no se toman en cuenta.

Cada regla tiene que estar en una línea separada.

Sintaxis:

```
<daemon_list>: <client_list>[: <shell_command> ]
```

Cada una de estas opciones se refiere a una parte específica de la regla:

- daemon_list: Es una colección de uno o varios nombres de procesos o de comodines especiales separados por espacios en blanco.

- client_list: Uno o varios nombres de máquinas, direcciones de máquina, funciones o comodines separados por espacios en blanco que se usan cuando el nombre de un determinado proceso se adapta a un determinado servicio.

- shell_command: Es un componente opcional que indica la realización de algo en el caso de que se use una regla.

Las características son de bastante ayuda cuando se especifican qué grupos de clientes pueden o no tener acceso a un determinado servicio. Así al poner el punto "." al comienzo de una cadena, todas las máquinas que se encuentran en esa cadena deben cumplir dicha regla. De tal manera que .domain.com engloba tanto a system1.domain.com como a system2.domain.com.

El punto "." al final de una cadena tiene el mismo efecto, excepto que el sentido cambia, ya que ésto se usa sobre todo para las direcciones IP ya que una regla relativa a 192.168.0. se aplica a todo el bloque C de las direcciones IP. Las expresiones de las máscaras de red también se pueden usar como característica para controlar el acceso a un determinado grupo de direcciones IP.

También puede usar asteriscos (*) o interrogaciones (?) para seleccionar grupos enteros de nombres de máquinas o de direcciones IP siempre y cuando no las use en la misma cadena de las otras características.

Si la lista de los nombres de máquinas que tienen acceso a un determinado servicio es muy larga o es difícil de controlar en los archivos host.allow o hosts.deny, puede especificar el recorrido completo a un archivo (como /etc/telnet.hosts.deny). Este archivo contiene diferentes nombres de máquinas, direcciones de máquinas o características separados por espacios en blanco a los que desea conceder o denegar el acceso a dicho servicio.

El método también funciona para compartir las listas de control del acceso entre varios servicios ya que los cambios que se quieran realizar solamente se tienen que hacer en un archivo para que afecten a cada servicio.

Los siguientes comodines se usan en las reglas para el control del acceso en vez de usar determinados nombres de máquinas o grupos de nombres de máquinas:

- ALL: Para permitir a un cliente tener acceso a todos los servicios, se utiliza ALL en la sección de los demonios.

- LOCAL: Hace corresponder todos los nombres de máquinas que no contengan un punto ".".

- KNOWN: Hace corresponder todas las máquinas cuyos nombres y direcciones se conozcan o en el caso en que se conozca el usuario.

- UNKNOWN: Hace corresponder todas las máquinas cuyos nombres y direcciones sean desconocidas o en el caso en el que se desconozca el usuario.

- PARANOID: Hace corresponder todas las máquinas cuyo nombre no se corresponda con la dirección.

El script de instalación del LTSP agregará algunas entradas en el archivo /etc/hosts.allow para permitir que dhcp, tftp y portmap funcionen correctamente. Estas entradas dan por hecho que se están usando direcciones IP de la clase C 192.168.0.0.

Si se utilizan direcciones diferentes se deben sustituir adecuadamente. Como el archivo hosts.allow tiene prioridad sobre el archivo hosts.deny, en realidad los cambios más importantes se efectúan en el primero.

El archivo hosts.allow tiene la siguiente estructura:

```
#
# hosts.allow    This file describes the names of the hosts which are
#                allowed to use the local INET services, as decided
#                by the '/usr/sbin/tcpd' server.
#

## LTS-begin ##

#
# The lines between the 'LTS-begin' and the 'LTS-end' were added
# on: sab sep 27 12:34:50 CDT 2003 by the ltsp installation script.
# For more information, visit the ltsp homepage
# at http://www.ltsp.org
#

bootpd:   0.0.0.0
in.tftpd: 192.168.0.
portmap:  192.168.0.

## LTS-end ##
```

Tabla 3-6 Ejemplo de configuración del archivo hosts.allow

Y el archivo hosts.deny, la siguiente:

```
#
# hosts.deny    This file describes the names of the hosts which are
#               *not* allowed to use the local INET services, as decided
#               by the '/usr/sbin/tcpd' server.
#
# The portmap line is redundant, but it is left to remind you that
# the new secure portmap uses hosts.deny and hosts.allow.  In particular
# you should know that NFS uses portmap!
```

Tabla 3-7 Ejemplo de configuración del archivo lts.conf

Estos archivos, permiten o niegan respectivamente algún servicio a las direcciones especificadas (en este caso las terminales), en algunas ocasiones pueden definirse direcciones IP concretas y en otros rangos de direcciones; incluso pueden combinarse.

Pero hay que tener cuidado al manipular estos archivos, ya que forman parte del firewall del sistema.

3.2.5.4.1 Terminal 1

Para este equipo, la configuración del archivo hosts.allow debe ser:

```
ALL:    192.168.0.1
```

En el archivo hosts.deny no es necesario hacer alguna configuración para esta terminal.

3.2.5.4.2 Terminal 2

Para este equipo, la configuración del archivo hosts.allow debe ser:

```
ALL:    192.168.0.2
```

En el archivo hosts.deny no es necesario hacer alguna configuración para esta terminal.

3.2.5.4.3 Terminal 3

Para este equipo, la configuración del archivo hosts.allow debe ser:

```
ALL:    192.168.0.3
```

En el archivo hosts.deny no es necesario hacer alguna configuración para esta terminal.

3.2.5.5 Exports

Es sencillo configurar un sistema para compartir archivos y directorios usando NFS, cada sistema de archivos que se exporta a usuarios remotos vía NFS, así como los derechos de acceso relativos a ellos, es localizado en el archivo /etc/exports. Este archivo es leído por el comando exportfs que da a rpc.mountd y rpc.nfsd la información necesaria para permitir el montaje remoto de un sistema de archivos por una máquina autorizada. Por omisión, rpc.mountd desaprueba el montaje de todos los directorios, lo cual es una medida de seguridad muy adecuada. Si se desea permitir montar un directorio de NFS, debe ser exportado, para lograr esto se debe especificar en el archivo exports.

Cada línea define un directorio y los anfitriones a los que se les permite montarlo. Un nombre de anfitrión es usualmente un nombre de dominio pero puede contener adicionalmente los comodines "*" y "?" los cuales funcionan igual que en el Bourne Shell de Linux. Por ejemplo, lab*.foo.com coincide con lab01.foo.com así como laboratory.foo.com. El anfitrión puede ser especificado también usando un rango de direcciones IP en la forma dirección/máscara de red.

Cuando se verifica a un anfitrión cliente contra el archivo exports, rpc.mountd busca el nombre del anfitrión cliente usando la llamada gethostbyaddr. Con DNS, esta llamada devuelve el nombre canónico del anfitrión cliente, así debe asegurarse de no usar alias en exports. En un entorno NIS el nombre devuelto es la primera coincidencia de la base de datos de anfitriones, y sin DNS o NIS, el nombre devuelto es el primer nombre de anfitrión encontrado en el archivo hosts que coincida con la dirección del cliente.

El nombre del anfitrión es seguido por una lista opcional de parámetros, los cuales pueden ser:

- secure: Asegura que la conexión se haga desde un puerto origen reservado, por ejemplo, uno que sea menor que 1024. Este parámetro es puesto por omisión.

- insecure: Este parámetro revierte el efecto de secure.

- ro: Provoca que el montaje de NFS sea para sólo lectura. Este parámetro está activado por omisión.

- rw: Esta opción monta la jerarquía de archivos en lectura-escritura.

- root_squash: Esta característica de seguridad deniega a los superusuarios en los hosts especificados cualquier derecho de acceso especial mapeando las peticiones desde el uid 0 en el cliente al uid 65534 (es decir, -2) en el servidor. Este uid debe ser asociado con el usuario nobody.

- no_root_squash: Esta opción no mapea las peticiones desde uid 0. Este parámetro está habilitado por omisión, así los superusuarios tienen acceso de supervisor a los directorios exportados de su sistema.

- link_relative: Convierte los enlaces simbólicos absolutos (donde el contenido del enlace comienza con un slash) en enlaces relativos. Esta opción sólo tiene sentido cuando está montado el sistema de archivos entero de un anfitrión; por otra parte, algunos de los enlaces podrían apuntar a ninguna parte, o peor aún, a archivos que nunca debieran apuntar. Esta opción está habilitada de forma predeterminada.

- link_absolute: Esta opción deja todos los enlaces simbólicos existentes.

- map_identity: Esta opción le indica al servidor asumir que el cliente usa el mismo uid y gid que el servidor. Este parámetro está habilitado por omisión.

- map_daemon: Indica al servidor de NFS asumir que el cliente y el servidor no comparten el mismo espacio uid/gid, rpc.nfsd entonces construye una lista que mapea los IDs entre cliente y servidor preguntando al demonio rpc.ugidd del cliente.

- map_static: Esta opción le permite especificar el nombre de un archivo que contiene un mapa estático de uids. Por ejemplo, map_static=/etc/nfs/vlight.map especificaría el archivo /etc/nfs/vlight.map como un mapa de uid/gid.

- map_nis: Esta opción causa que el servidor de NIS haga un mapeado de uid y gid.

- anonuid y anongid: Estas opciones le permiten especificar el uid y el gid de la cuenta anónima. Esto es útil si tiene un volumen exportado para montajes públicos.

En esencia, éste archivo define que directorios serán compartidos a otras máquinas, en el caso del LTSP, definirá los directorios a los cuales podrán tener acceso las terminales gráficas.

La estructura del archivo creada por los scripts del LTSP es la siguiente:

```
## LTS-begin ##
#
# The lines between the 'LTS-begin' and the 'LTS-end' were added
# on: Sat sep 27 12:34:50 CDT 2003 by the ltsp installation script.
# For more information, visit the ltsp homepage
# at http://www.ltsp.org
#

/opt/ltsp/i386              192.168.0.0/255.255.255.0(ro,no_root_squash,sync)
/var/opt/ltsp/swapfiles     192.168.0.0/255.255.255.0(rw,no_root_squash,async)

#
# The following entries need to be uncommented if you want
# Local App support in ltsp
#
#/home              192.168.0.0/255.255.255.0(rw,no_root_squash,sync)

## LTS-end ##
```

Tabla 3-8 Ejemplo de configuración del archivo exports

3.2.5.6 lts.conf

Este archivo contiene las configuraciones necesarias para crear el entorno LTSP; permite especificar los parámetros que las terminales requieren para poder funcionar correctamente. Este archivo tiene una sintaxis simple, que consiste de múltiples secciones. Hay una sección por defecto, llamada [default] y hay secciones adicionales para terminales individuales. Si todas las terminales son idénticas, entonces se deben especificar todas las opciones bajo la sección [Default]. Las terminales pueden ser identificadas por nombre de host, dirección IP o dirección MAC. A continuación se describen los parámetros principales que acepta el archivo lts.conf:

A) Opciones generales

- Comentarios: Los comentarios comienzan con un símbolo de número ('#') y continúan hasta el final de la línea.

- LTSP_BASEDIR: Indica dónde está ubicado el sistema de archivos raíz del LTSP; por defecto es en /opt/ltsp/.

- SERVER: Este es el servidor utilizado por XDM_SERVER, TELNET_HOST, XFS_SERVER y SYSLOG_HOST, si alguno de los anteriores no está especificado explícitamente. Si se tiene una computadora que está actuando como el servidor de

todo, entonces se puede especificar su dirección IP aquí y omitir el resto. Si ningún valor es especificado, se toma por defecto la dirección 192.168.0.254.

- SYSLOG_HOST: Si se requiere enviar los mensajes de sistema a una máquina distinta al servidor por defecto, entonces se tiene que especificar dicha máquina aquí. Si ningún parámetro es especificado se utilizará la entrada de SERVER.

- NFS_SERVER: Especifica la dirección IP del servidor desde donde el sistema de archivos /home es montado. Por defecto se utiliza la entrada SERVER.

- USE_NFS_SWAP: Esta opción habilita el uso de memoria SWAP sobre NFS. Por defecto es N.

- SWAPFILE_SIZE: Permite controlar el tamaño del archivo de intercambio. El tamaño por defecto es 64MB.

- SWAP_SERVER: El archivo de intercambio puede existir en cualquier servidor de la red. Se puede especificar la dirección IP de dicho servidor. Por defecto se toma el valor de NFS_SERVER.

- NFS_SWAPDIR: Especifica el directorio que es exportado vía NFS. Por defecto es /var/opt/ltsp/swapfiles. Es necesario que dicho directorio esté registrado en el archivo /etc/exports.

- TELNET_HOST: Si la terminal es configurada para tener una sesión en modo texto, entonces el valor de este parámetro será usado como el host donde se haga la sesión de telnet. Si ningún valor es especificado se utilizará la entrada de SERVER.

- DNS_SERVER: Usada para crear el archivo resolv.conf.

- SEARCH_DOMAIN: Usada para crear el archivo resolv.conf.

MODULE_01 hasta MODULE_10

10 módulos de kernel pueden ser cargados mediante estas entradas. Por ejemplo:

```
MODULE_01 = uart401.o
MODULE_02 = sb.o io=0x220 irq=5 dma=1
MODULE_03 = opl3.o
```

Si el valor de este parámetro es una ruta absoluta, entonces el comando insmod será utilizado. De otro modo, se utilizará modprobe.

- RAMDISK_SIZE: Cuando la terminal se inicia, crea un disco RAM y lo monta en el directorio /tmp. Se puede controlar el tamaño de este sistema de archivos con este parámetro. Se debe especificar las unidades en kilobytes (1024 bytes). Por ejemplo para crear un disco RAM de 2 MB se debe especificar RAMDISK_SIZE=2048.

- RCFILE_01 hasta RCFILE_10: Algunos scripts adicionales pueden ser ejecutados por rc.local. Se debe colocar el script en el directorio /etc/rc.d y especificar el nombre en alguna de estas entradas.

- SOUND: Si el paquete LTSP Sound está instalado, se necesita poner el valor 'Y' en esta entrada y ejecutar el script rc.sound para configurar la tarjeta de sonido y el demonio. Por defecto es N.

- RUNLEVEL: El runlevel determina el modo en el cual la terminal estará corriendo. Los siguientes runlevels son los soportados:

  3: Iniciará un shell. Útil para depuración de la terminal.
  4: Correrá una o más sesiones de telnet en el TELNET_HOST.
  5: Modo GUI. Arrancará X Window, y enviará una petición XDMCP al servidor, quien devolverá una pantalla de login a la terminal. Se necesita tener un Display Manager corriendo en el servidor, como XDM, GDM o KDM.

- TELNET_SESSIONS: Indica cuántas sesiones de telnet se pueden correr. Cada sesión estará en una diferente pantalla virtual, a las que se puede acceder con las teclas ALT-F1 a ALT-F9. El valor por defecto es 2.

B) Opciones de X Window

- XDM_SERVER: Si se requiere que XDM corra en una máquina que no sea el servidor por defecto, entonces se debe especificar aquí. Si este parámetro no es especificado, entonces se usará la entrada en SERVER.

- XSERVER: Esta entrada define qué servidor X correrá la terminal. Para placas PCI y AGP, este parámetro no debería ser necesario. El script rc.local debería autodetectar la tarjeta. También se puede activar auto a fin de indicar que se tratará de autodetectar el video. Para placas ISA, o para especificar un servidor X en particular, se puede poner el nombre del driver o servidor X. Si el valor comienza con 'XF86_', entonces XFree 3.3.6 será utilizado, sino, XFree 4.1.x lo será. El valor por defecto es auto.

- X_MODE_0 hasta X_MODE_2 : Hasta 3 resoluciones pueden ser configuradas para la terminal. Esta entrada puede tomar dos tipos diferentes de valores. Puede ser tanto una resolución como un modeline completo:

X_MODE_0 = 800x600 o bien

X_MODE_0 = 800x600 60.75 800 864 928 1088 600 616 621 657 -HSync –Vsync

Si ninguna de las entradas X_MODE_x es especificada, entonces se utilizarán los modelines por defecto, y las resoluciones serán de 1024x768, 800x600 y 640x480. Si una o más entradas X_MODE_x es especificada, entonces no se tendrán en consideración los modelines por defecto.

- X_MOUSE_PROTOCOL: Cualquier valor que funcione con el protocolo de puntero XFree86 puede ser especificado. Los valores típicos incluyen 'Microsoft' y 'PS/2'. El valor por defecto es 'PS/2'.

- X_MOUSE_DEVICE: Este es el dispositivo al cual el mouse está conectado. Si es un mouse serial, debería ser un puerto serie, como /dev/ttyS0 o /dev/ttyS1. Si es un mouse PS/2, este valor debería ser /dev/psaux. El valor por defecto es /dev/psaux.

- X_MOUSE_RESOLUTION: Este es el valor 'Resolution' que se encuentra en cualquier archivo XF86Config. Un valor típico para un mouse serial es 50 y para uno PS/2 es 400. El valor por defecto es 400.

- X_BUTTONS: Indica al sistema cuántos botones tiene el mouse, usualmente son 2 o 3 y el valor por defecto es 3.

- X_MOUSE_EMULATE3BTN: Esto le indica al servidor X que debe emular el tercer botón del mouse cuando se presionen simultáneamente los botones izquierdo y derecho. El valor por defecto es N.

- X_MOUSE_BAUD: Para ratones seriales, define la tasa de baudios. El valor por defecto es 1200.

- X_COLOR_DEPTH: Este es el número de bits a utilizar para la profundidad del color. Los valores posibles son 8, 15, 16, 24 y 32. 8 bits darán 256 colores, 16 - 65536, 24 - 16 millones y 32 - 4,2 billones de colores. No todos los servidores X soportan estos valores. El valor por defecto es 16.

- USE_XFS: Se tiene la opción de correr un Servidor de Fuentes X (XFS, X Font Server) o bien leer las fuentes vía NFS. El servidor de fuentes debería proveer un camino simple para mantener todas las fuentes en un solo lugar, pero hay algunos problemas cuando el número de terminales supera las 40. Los 2 valores en esta opción son Y o N, el valor por defecto es N. Si se desea usar un Servidor de Fuentes, entonces se debe utilizar la entrada XFS_SERVER para especificar su dirección IP.

- XFS_SERVER: Si se está usando un Servidor de Fuentes, se debe especificar su dirección IP. Si no está especificado, entonces se usará por defecto la entrada especificada en SERVER.

- X_HORZSYNC: Define la sincronización horizontal del monitor y el valor por defecto es "31-62".

- X_VERTREFRESH: Define con que frecuencia se refresca verticalmente el monitor. Por defecto es "55-90".

- XF86CONFIG_FILE: Si se desea tener un archivo propio de configuración XF86Config, se puede especificar en esta opción. Sólo se tiene que colocar en el directorio /opt/ltsp/i386/etc. No importa su nombre, siempre y cuando dicho nombre sea indicado en éste parámetro. Por ejemplo:

- XF86CONFIG_FILE = XF86Config.ws004: Los scripts de configuración del LTSP crean este archivo el cual tiene la siguiente estructura:

```
    # Keyboards
    XkbSymboles     = "us(pc101)"
    XkbModel        = "pc101"
    XkbLayout       = "us"

    USE_XFS         = N
    LOCAL_APPS      = N
    RUNLEVEL        = 5

    # uncomment the following line to enable floppy support
    #RCFILE_01      = floppyd

    # uncomment the following line to enable USB support
    #RCFILE_02      = usb

    # enable sound by default
    SOUND           = Y

    # default sound volume
    VOLUME          = 75

    ### For ISA sound cards, you have to specify the module to use:
    # SMODULE_01    = sb io=0x220 irq=5 dma=1
#-----------------------------------------------------------------------
# Below are sample mode lines for a variety of vertical refresh rates and
# resolutions.  They are used to define the default client screen resolution.
# Some lines may not work with a given monitor and video chipset.
# To avoid damaging a monitor and video card, only specify mode lines that
# your client's hardware can support.
# Uncomment only *one* of the following X_MODE_0 lines at a time, or add one
# of your own.
```

```
# 60 Hz Resolutions
#  X_MODE_0 = 640x480  25.175  640  656  752  800   480 490 492 525  -hsync -vsync
#  X_MODE_0 = 800x600  40      800  840  968 1056   600 601 605 628  +hsync +vsync
#  X_MODE_0 = 1024x768 65     1024 1048 1184 1344   768 771 777 806  -hsync -vsync

# 70 Hz Resolutions (Use instead of 72 Hz for 1024x768)
#  X_MODE_0 = 1024x768 75     1024 1048 1184 1328   768 771 777 806  -hsync -vsync

# 72 Hz Resolutions
#  X_MODE_0 = 640x480  31.5    640  664  704  832   480 489 492 520  -hsync -vsync
#  X_MODE_0 = 800x600  50      800  856  976 1040   600 637 643 666  +hsync +vsync
#  X_MODE_0 = 1024x768 75     1024 1048 1192 1296   768 771 777 806  -hsync -vsync

# 75 Hz Resolutions
#  X_MODE_0 = 800x600  49.5    800  816  896 1056   600 601 604 625  +hsync +vsync

# 85 Hz Resolutions
#  X_MODE_0 = 800x600  60.75   800  864  928 1088   600 616 621 657  -hsync -vsync

#----------------------------------------------------------------------
#
# Example of specifying X settings for a workstation
#
[ws001]
    XSERVER         = auto
    LOCAL_APPS      = N
    USE_NFS_SWAP    = N
    SWAPFILE_SIZE   = 48m
    RUNLEVEL        = 5

[ws002]
    XSERVER         = XF86_SVGA
    LOCAL_APPS      = N
    USE_NFS_SWAP    = N
    SWAPFILE_SIZE   = 64m
    RUNLEVEL        = 3

#
# ws004 is my virtual workstation running in a VMware session
#
[ws004]
    DNS_SERVER      = 192.168.0.254
    XSERVER         = auto
    X4_BUSID        = "PCI:0:15:0"
    X_MODE_0        = 800x600
    LOCAL_APPS      = N
    USE_NFS_SWAP    = N
    SWAPFILE_SIZE   = 64m
    RUNLEVEL        = 5

#----------------------------------------------------------------------
#
# Example of a workstation configured to load some modules
```

```
#
#[ws001]
#   MODULE_01      = agpgart.o          # This is for i810 video
#   MODULE_02      = uart401.o
#   MODULE_03      = sb.o io=0x220 irq=5 dma=1
#   MODULE_04      = opl3.o

#-----------------------------------------------------------------------
#
# Example of ws001 configured for local apps
#
#[ws001]
#   LOCAL_APPS     = Y
#   LOCAL_WM       = Y
#   NIS_DOMAIN     = ltsp
#   NIS_SERVER     = 192.168.0.254

#-----------------------------------------------------------------------
#
# Example of a serial printer attached to /dev/ttyS1 on workstation ws001
#
#[ws001]
#   PRINTER_0_DEVICE  = /dev/ttyS1
#   PRINTER_0_TYPE    = S          # P-Parallel, S-Serial
#   PRINTER_0_PORT    = 9100       # tcp/ip port: defaults to 9100
#   PRINTER_0_SPEED   = 9600       # baud rate: defaults to 9600
#   PRINTER_0_FLOWCTRL = S         # Flow control: S-Software (XON/XOFF),
#                        # H-Hardware (CTS/RTS)
#   PRINTER_0_PARITY  = N          # Parity: N-None, E-Even, O-Odd
#                        # (defaults to 'N')
#   PRINTER_0_DATABITS = 8         # Databits: 5,6,7,8 (defaults to 8)
```

Tabla 3-9 Ejemplo de configuración del archivo lts.conf

3.2.5.6.1 Terminal 1

Para este equipo, la configuración del archivo lts.conf debe ser:

```
[ws001]
        XSERVER             = Xvesa
        LOCAL_APPS          = N
        USE_NFS_SWAP        = N
        SWAPFILE_SIZE       = 48m
        RUNLEVEL            = 5
        X_HORZSYNC          = "30.9-49.0"
        X_VERTREFRESH       = "50.0-100.0"
        X_MODE_0            = 800x600
```

3.2.5.6.2 Terminal 2

Para este equipo, la configuración del archivo lts.conf debe ser:

```
[ws002]
    XSERVER          = Xvesa
    LOCAL_APPS       = N
    USE_NFS_SWAP     = N
    SWAPFILE_SIZE    = 48m
    RUNLEVEL         = 5
    X_HORZSYNC       = "30.9-49.0"
    X_VERTREFRESH    = "50.0-100.0"
    X_MODE_0         = 800x600
```

3.2.5.6.3 Terminal 3

Para este equipo, la configuración del archivo lts.conf debe ser:

```
[ws003]
    XSERVER          = Xvesa
    LOCAL_APPS       = N
    USE_NFS_SWAP     = N
    SWAPFILE_SIZE    = 48m
    RUNLEVEL         = 5
    X_MODE_0         = 800x600
```

3.3 Terminales

Una vez que se haya terminado de instalar y configurar todos los paquetes necesarios en el servidor, el siguiente paso es crear una imagen de sistema operativo para que las terminales puedan arrancar. Donde las posibilidades de arranque podrían ser las siguientes:

- Arranque desde la tarjeta de red: Las tarjetas necesitan un chip EPROM para grabar en él la imagen de arranque. Arranque desde la propia BIOS si se incorpora PXE (Preboot-eXecution Environment).

- Arranque desde disquete: La imagen de arranque se graba en un disquete 3 ½ " de alta densidad. Esta es la solución más económica y sencilla.

- Arranque desde el disco duro: Configurando el archivo Config.sys y Autoexec.bat en WINDOWS.

3.3.1 Creación del disquete de arranque

Es importante mencionar que no basta con crear una imagen de arranque común, la diferencia radica en que ésta deberá estar etiquetada especialmente para que el kernel permita arrancar la terminal vía red.

La forma más común es bajar la imagen de arranque y copiarla en un disquete[13]. Se deberá elegir la opción Floppy Bootable Rom Image, lo cual generará una imagen que se puede escribir en un disquete con los comandos dd o cat, desde Linux; o rawrite en MS-DOS.

Este disquete primero intentará detectar la tarjeta de red y luego iniciará el proceso de arranque como si el código estuviera en una eprom desde la misma tarjeta de red. Por ejemplo, para generar el disquete de arranque para la tarjeta MX98715 de la terminal 3, se tiene que hacer lo siguiente:

- Ir al sitio y hacer click en la liga de la versión 5.0.7.
- Seleccionar el tipo de tarjeta de red, en este caso es "mx98715".
- Elegir el formato de salida Floppy Bootable ROM Image (.lzdsk).
- Hacer click en "Get ROM" para generar la imagen de arranque. Se creará el siguiente archivo:

```
eb-5[1].0.7-mx98715.lzdsk
```

Para crear el disco de arranque no basta con copiar la imagen al disquete, es necesario que éste sea booteable para esto se utiliza cualquiera de los siguientes comandos:

En Linux se utiliza:

```
cat eb-5[1].0.7-mx98715.lzdsk > /dev/fd0
```

Esto copiará la imagen en el primer sector del disquete.

[13] Página del sitio para descargar la imagen de arranque http://www.rom-o-matic.net

O desde MS-DOS, usando el comando rawrite, el cual viene incluido en el primer disco de instalación de Fedora Core, en la carpeta DOSUTILS.

Se escribe el comando:

```
D:\DOSUTILS\rawrite
```

Se pedirá la ruta de la imagen de arranque:

```
C:\eb-5[1].0.7-mx98715.lzdsk
```

Y después la ruta donde se escribirá la imagen de arranque, en este caso es el disquete:

```
A:\
```

Este comando no acepta archivos con extensiones mayores a tres caracteres, por lo tanto se tendrá que cambiar la extensión del archivo, en este caso será .img.

```
C:\ren eb-5[1].0.7-mx98715.lzdsk   mx98715.img
```

Y se tendrá que repetir el proceso con el comando rawrite hasta que se haya logrado grabar correctamente la imagen en el disquete.

Después de crear el disco de arranque, será necesario probar que éste reconoce la tarjeta de red de la terminal, lo cual se puede comprobar si al bootear la terminal con el disquete se muestra la MAC address de la tarjeta y aparece un mensaje que indica que se está buscando un servidor DHCP.

3.3.2 Arranque por medio del disco duro

Otra alternativa para el arranque de las terminales es la creación de un menú en Windows en el disco duro local, lo cual permite iniciar Linux sin la necesidad de utilizar un disquete de arranque y a su vez iniciar Windows normalmente.

Es importante mencionar que esté método solo funciona en Windows 95 y 98, ya que en versiones posteriores el soporte para el menú del archivo config.sys fue

deshabilitado. Las terminales 1 y 2 tienen instalado Windows 95 por lo que utilizarán este método, en cambio la terminal 3 tiene instalado Windows Millenium, por lo que se utilizará un disquete de arranque.

Para crear el menú se requiere hacer lo siguiente:

- Ir al sitio[14] y hacer click en la liga de la versión 5.0.7.
- Seleccionar el tipo de tarjeta de red, en este caso es "3c509" para las terminales 1 y 2.
- Elegir el formato de salida DOS Executable ROM Image (.com).
- Hacer click en "Configure" para definir que en el proceso de arranque se muestre una barra giratoria en lugar de una sucesión de puntos.
- Hacer click en "Get ROM" para generar la imagen de arranque. Se creará el siguiente archivo:

```
eb-5[1].0.7-3c509.com
```

Se debe renombrar el archivo generado con un nombre descriptivo, no mayor a 8 caracteres. Esto se puede hacer directamente en el explorador de Windows o en la línea de comandos de MS-DOS.

```
C:\ren  eb-5[1].0.7-3c509.com   3c509.com
```

Posteriormente se debe crear un directorio directamente en C:\ llamado ltsp, donde se colocará el archivo 3c509.com.

```
C:\md ltsp
C:\cp 3c509.com  ltsp\
```

Se tiene que editar el archivo config.sys y agregar lo siguiente al inicio:

[14] Página del sitio para crear al menú de inicio http://www.rom-o-matic.net

```
[Menu]
MenuItem=windows,Microsoft Windows 95/98
MenuItem=linux,Linux Terminal Server Project
MenuDefault=windows,10

[linux]
DOS=NOAUTO

[windows]
```

Esto creará el menú con las opciones de inicio Windows y Linux.

También se tiene que editar el archivo autoexec.bat agregando lo siguiente al inicio:

```
@ECHO OFF
GOTO %CONFIG%

:linux
C:\ltsp\3c509.com

:windows
```

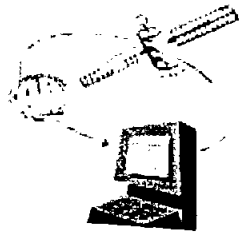
Esto permitirá realmente ejecutar la imagen de arranque una vez que se haya seleccionado la opción Linux en el menú de inicio.

Hasta éste punto preparamos el equipo para instalar adecuadamente el sistema operativo así como los diferentes componentes que intervienen en la implementación; fueron instalados los archivos de sistema y aprendimos a crear el disco de arranque. Ahora estamos listos para iniciar todo el sistema, analizaremos el inicio de una terminal para lograr detectar problemas y con lo que hemos visto en los capítulos anteriores podremos instalar sin problemas el software adicional así como la configuración a internet.

PUESTA EN MARCHA

OBJETIVO ESPECÍFICO

Aplicar los conocimientos teóricos y prácticos para garantizar el buen funcionamiento del servidor de terminales gráficas en el momento de iniciar todo el sistema. Entender el proceso de cómo inicia una terminal para poder detectar y darle la solución más óptima a los posibles problemas.



4

INTRODUCCIÓN

Hasta este punto se han instalado los paquetes del LTSP y del software necesario, también se han realizado las configuraciones correspondientes a los archivos de sistema; sin embargo falta comprobar que efectivamente cada una de las terminales inicien una sesión gráfica debido a que estas configuraciones no garantizan que dichas terminales funcionen correctamente, para ello se cuenta con un apartado de detección y solución de problemas más comunes.

Una vez que han sido solucionados los problemas y se han iniciado las terminales gráficamente estarán listas para ser conectadas en red de forma local y a su vez tendrán acceso a la red global.

Tomando en cuenta que en el laboratorio se utiliza software comercial de electrónica, se hizo una recopilación de software de este tipo, pero que puede correr bajo Linux y con la ventaja de que es libre.

4.1 Teoría de operación

El arranque de una terminal gráfica aplicando el LTSP involucra muchos pasos, es importante conocer y comprender de forma general cada uno de ellos; ya que si en algún momento surgen problemas, estos podrían solucionarse más rápidamente, debido a que se puede deducir el origen de ellos.

El proceso de arranque de una terminal aplicando el LTSP es el siguiente:

- Cuando se enciende la terminal, ésta busca entre los dispositivos de almacenamiento disponibles un sistema operativo que iniciar; cuando detecta el código de arranque, éste se empezará a ejecutar.

- El código de inicio tratará de detectar una tarjeta de red y una vez que lo hace, la inicializa.

- Se hará una petición de broadcast buscando un servidor dhcp en la red local, esta petición incluirá la dirección MAC de la tarjeta de red.

- El proceso inetd del servidor verá la petición de broadcast e invocará al demonio dhcpd para responderla.

- El demonio dhcpd leerá el archivo de configuración, dhcpd.conf, y tratará de localizar la entrada que corresponda con la dirección MAC de la terminal. Al ser encontrada, la pondrá en un paquete de respuesta y se lo enviará a la terminal que pidió la información. Ciertos parámetros serán incluidos en este paquete, siendo los más importantes:

    - Dirección IP asignada a la terminal.
    - Máscara de subred de la red de área local.
    - Directorio home del archivo de inicio.
    - Nombre del kernel a bajar.

- El código de inicio recibirá el paquete de respuesta dhcp y configurará la interfaz TCP/IP en la tarjeta de red con los parámetros proporcionados.

- El código de inicio enviará una petición TFTP al servidor para empezar a bajar el kernel.

- Una vez que el kernel fue completamente bajado por la terminal, éste tomará el control.

- El kernel empezará a ejecutarse, inicializando todo el sistema y sus periféricos.

- El código de inicio le puede pasar información al kernel y lo hace construyendo una línea de comandos, similar a la manera en que LILO o GRUB le pasa parámetros.

- Si el kernel necesita hacer una petición dhcp, el servidor responderá con otro paquete, conteniendo la información que el kernel necesita para continuar. Los parámetros importantes en esta respuesta son:

    - Dirección IP asignada a la terminal.
    - Máscara de subred de la red de área local.
    - El directorio raíz a ser montado vía NFS.
    - La puerta de enlace (gateway).
    - El servidor de nombres (DNS).
    - El nombre de host de la terminal.

- El directorio raíz de la terminal será montado vía NFS. Este sistema de archivos será de sólo lectura; porque se pueden tener varias terminales montando el mismo sistema de archivos, y con esto se evita que en alguna de ellas se modifiquen los contenidos del mismo.

- En este punto, el control será pasado del kernel al proceso init.

- Init leerá el archivo inittab y empezará a configurar el ambiente.

- Uno de los primeros elementos de inittab es el archivo rc.local, que correrá mientras la terminal está en el estado 'sysinit'.

- El script en rc.local creará un disco RAM de 1 MB que contendrá todas las cosas que necesitan ser escritas o modificadas de alguna manera.

- Este disco RAM será montado en el directorio /tmp. Cualquier archivo que necesite ser escrito estará en el directorio /tmp, con un enlace simbólico apuntando hacia él.

- El directorio /proc es montado.

- La interfaz de loopback es configurada.

- Varios directorios son creados dentro de /tmp para que contengan algunos de los archivos transitorios que son necesarios mientras el sistema está corriendo. Algunos de ellos son:

    - /tmp/compiled
    - /tmp/var
    - /tmp/var/run

- /tmp/var/log
- /tmp/var/lock
- /tmp/var/lock/subsys

- El archivo XF86Config es generado y está basado en las entradas del archivo lts.conf. Aquí es donde la información acerca del tipo de mouse, y otros parámetros para el servidor X son combinados para crear el archivo de configuración.

- El script start_ws es creado, éste es responsable de iniciar el servidor X y cuando lo inicia, es dirigido para enviar peticiones XDMCP al servidor XDM. Esto funciona basado en la información encontrada en el archivo lts.conf.

- El archivo syslog.conf es creado, éste archivo contiene información que le indica al demonio syslogd a qué host de la red le deberá enviar la información de log. Este host de syslog está especificado en el archivo lts.conf. Hay un enlace simbólico llamado /etc/syslog.conf que apunta a /tmp/syslog.conf.

- El demonio syslogd se inicia, usando el archivo de configuración recientemente creado.

- El control es pasado otra vez a init y buscará en la entrada initdefault para saber a cuál runlevel entrar, por lo general el runlevel por defecto es 2.

- El runlevel 2 hará que se ejecute el script set_runlevel, que leerá el archivo lts.conf para saber en qué modo correr la terminal.

- Los niveles de ejecución estándar de LTSP son 3, 4 y 5.

  - El nivel 3 iniciará en un shell en la terminal.
  - El nivel 4 ejecutará una o más sesiones de telnet en el servidor. Esto es ideal para remplazar viejas terminales de tipo serial.
  - El nivel 5 entrará en el modo GUI, levantando el cliente X y haciendo una petición XDMCP al servidor, quien mandará una ventana de diálogo de inicio para permitir al usuario autenticarse. Hace falta un Display Manager corriendo en el servidor, como XDM, GDM o KDM.

4.2 Iniciando las terminales

La estrategia que se adoptó fue probar de manera independiente cada una de las terminales, para lo cual es necesario conectarlas directamente al servidor por medio de un cable cruzado; la ventaja de hacerlo de esta forma es que se tiene un mejor control en cuanto a la posible detección y solución de problemas.

Una vez que se haya comprobado que todas las terminales inician correctamente el sistema operativo y una respectiva sesión gráfica, se podrá implementar realmente la red, es decir; las terminales se conectarán al switch y éste a su vez al servidor, para lo cual será necesario utilizar cable directo en lugar de cable cruzado.

Asumiendo que las configuraciones del servidor fueron las adecuadas, al iniciar una sesión será desplegada la siguiente pantalla:

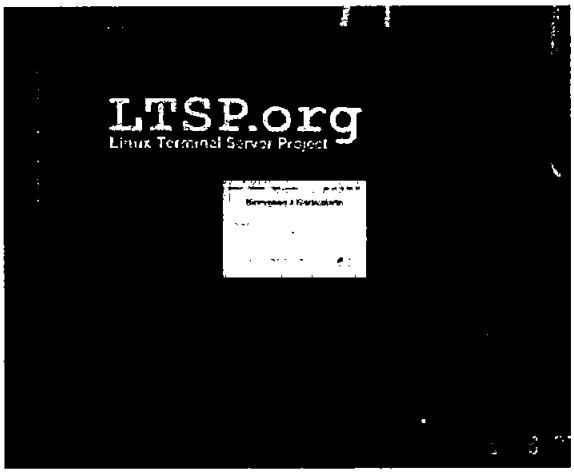


Figura 4-1 Pantalla de bienvenida del LTSP

Debido a su importancia, se describirá en la siguiente sección los problemas que se detectaron así como otros posibles errores en la implementación del LTSP, además la forma en que pueden ser solucionados.

4.3 Detección y solución de problemas

Los scripts de inicialización del LTSP hacen un gran trabajo al realizar ciertas configuraciones en los archivos necesarios, pero debido a la gran diversidad de hardware disponible para las terminales, es poco probable que éstas inicien completamente y de manera correcta la primera vez; es por eso que se debe comenzar con un proceso de detección y solución de problemas.

Conforme se va teniendo más experiencia en el proyecto LTSP, resulta más sencillo detectar los posibles errores y determinar la solución apropiada en cada caso; para ello es necesario observar hasta que punto ha llegado la terminal en su proceso de inicio.

A continuación se presentan los problemas más comunes, la manera de cómo detectarlos y sus posibles soluciones.

4.3.1 Conexiones

Es probable que la terminal no esté conectada apropiadamente a la red local, sobre todo cuando apenas se está implementando dicha red y ni siquiera se ha probado la conectividad entre las máquinas.

Para verificar esto, hay que observar que las luces de la tarjeta de red estén encendidas así como para todas las conexiones de red. Esto garantiza que al menos existe la conectividad entre la terminal y el dispositivo de la red local correspondiente.

En algunas ocasiones por cuestiones de prueba, es conveniente trabajar con la terminal conectada directamente al servidor; para ello es necesario que entre la terminal y el servidor (sin hacer uso de un hub o switch) se esté utilizando un cable cruzado.

Si se está utilizando un hub o switch, entonces hay que asegurarse que se está usando un cable directo, tanto entre las terminales y el hub, como entre el hub y el servidor.

En el capítulo 2 se muestra la configuración de los pines para construir un cable directo y un cable cruzado.

4.3.2 Imagen de arranque

La imagen de arranque debe reconocer la tarjeta de red de la terminal para poderla inicializar y configurar correctamente; si esta imagen no corresponde a la tarjeta de red, entonces no servirá para inicializar dicha terminal.

Cuando se inicia la terminal por medio de la imagen de arranque, se debe observar una salida como la siguiente:

```
loaded ROM segment 0x0800 length 0x4000 reloc 0x9400
Etherboot 5.0.1 (GPL) Tagged ELF for [LANCE/PCI]
Found AMD Lance/PCI at 0x1000, ROM address 0x0000
Probing...[LANCE/PCI] PCnet/PCI-II 79C970A base 0x1000, addr 00:50:56:81:00:01
Searching for server (DHCP)...
<sleep>
```

Tabla 4-1 Mensaje de salida cuando la imagen de arranque es apropiada

Para identificar si la imagen de arranque es la apropiada, ésta debe mostrar la dirección MAC de la tarjeta de red, sin embargo esto no garantiza que funcione completamente.

Si, por el contrario, se obtiene una salida como la siguiente, entonces es probable que la imagen de arranque generada no sea la apropiada para la tarjeta de red ya que no detecta ni muestra la dirección MAC.

```
ROM segment 0x0800 length 0x8000 reloc 0x9400
Etherboot 5.0.2 (GPL) Tagged ELF for [Tulip]
Probing...[Tulip]No adapter found
<sleep>
<abort>
```

Tabla 4-2 Mensaje de salida al iniciar la terminal con la imagen de arranque

Si no se muestran estos mensajes indicando que Etherboot a iniciado, entonces puede ser que el disquete esté dañado o que se haya grabado la imagen de arranque de manera incorrecta.

En el capítulo 3 se indica la manera de generar la imagen de arranque, así como la forma de grabarla en el disquete.

4.3.3 Problemas con el servicio DHCP

Una vez que la tarjeta de red es inicializada, el código de la imagen de arranque enviará una petición de broadcast a través de la red local, buscando algún servidor DHCP. Si la terminal obtiene una respuesta válida de parte del servidor DHCP, deberá configurar la tarjeta de red.

Si el servidor DHCP está funcionando correctamente, éste responderá a la terminal enviándole un paquete con la información que está definida en su archivo de configuración (dhcpd.conf) y que corresponde a la terminal que envió la petición.

```
ROM segment 0x0800 length 0x4000 reloc 0x9400
Etherboot 5.0.1 (GPL) Tagged ELF for [LANCE/PCI]
Found AMD Lance/PCI at 0x1000, ROM address 0x0000
Probing...[LANCE/PCI] PCnet/PCI-II 79C970A base 0x1000, addr 00:50:56:81:00:01
Searching for server (DHCP)...
<sleep>
Me: 192.168.0.1, Server: 192.168.0.254, Gateway 192.168.0.254
```

Tabla 4-3 Mensaje de que el servidor DHCP funciona correctamente

Si se muestra la línea que comienza con 'Me:', seguida por una dirección IP, entonces es un hecho que DHCP está funcionando correctamente. En ésta línea se mostrará la dirección IP asignada a la terminal, y las direcciones IP del servidor y del gateway.

Si por el contrario se muestra el siguiente mensaje en la terminal y no se obtiene respuesta por parte del servidor, es probable que haya un problema con el servicio DHCP.

```
Searching for server (DHCP)...
<sleep> <sleep>
```

Tabla 4-4 Mensaje de que quizá no funciona adecuadamente el servicio DHCP

Hay que notar que el mensaje <sleep> se puede mostrar por algunos segundos, debido al tráfico en la red, o porque por alguna razón el servidor estaba muy ocupado y no había podido atender la petición de la terminal; sin embargo no es normal que el servidor tarde mucho tiempo en dar una respuesta.

En el caso de que el servidor DHCP no responda, se tiene que averiguar las posibles causas; entre las cuales se encuentran las siguientes.

4.3.3.1 El demonio dhcpd

Para determinar si el demonio dhcpd está corriendo en el servidor se pueden ejecutar algunos comandos, e interpretar su salida; dhcpd normalmente corre en modo background escuchando en el puerto udp 67.

Se debe ejecutar el comando netstat para ver si hay algo escuchando en dicho puerto:

```
netstat -an | grep ":67 "
```

Se debe mostrar una salida similar a la siguiente:

```
udp    0      0      0.0.0.0:67      0.0.0.0:*
```

Tabla 4-5 Mensaje para saber si el demonio dhcpd esta corriendo en el servidor

La cuarta columna contiene la dirección IP y el puerto, separadas por dos puntos ":". Una dirección con solo ceros "0.0.0.0" indica que se está escuchando en todas las

interfases. Esto significa que se pueden tener varias tarjetas de red en el servidor; una interfaz eth0, otra eth1, etc., y que dhcpd esté escuchando en todas ellas.

Aunque se haya mostrado que existe un servicio escuchando en el puerto udp 67, esto no significa que efectivamente dhcpd sea el que lo esté haciendo; ya que BOOTP el cual es un servicio similar a DHCP, podría ser el que estuviera ocupando el puerto udp 67.

Actualmente en la mayoría de las distribuciones Linux solo se incluye el servicio DHCP, pero para asegurarse que el demonio esté ejecutándose realmente en el servidor, se tiene que teclear el siguiente comando:

```
ps aux | grep dhcpd
```

La salida es parecida a la siguiente:

```
root 23814 0.0 0.3 1676 820 ? S 15:13 0:00 /usr/sbin/dhcpd
root 23834 0.0 0.2 1552 600 pts/0 S 15:52 0:00 grep dhcp
```

Tabla 4-6 Mensaje para asegurarse que el demonio dhcpd esté ejecutándose

En realidad la primera línea es la que muestra que dhcpd está corriendo, la segunda es tan sólo el comando grep que se utilizó para hacer la búsqueda en la salida del comando ps.

Si no se muestra ninguna línea indicando que dhcpd está corriendo, entonces se necesita verificar que el servidor esté configurado para el runlevel 5 y que dhcpd esté listo para iniciar en dicho nivel. En sistemas basados en Fedora Core, se puede ejecutar el programa ntsysv para asegurar que el servicio DHCP esté seleccionado en el nivel de ejecución apropiado.

Existe una alternativa al comando ps, para saber si se está ejecutando dhcpd; se debe teclear lo siguiente:

```
service dhcpd status
```

Este comando mostrará el estado del servicio (si está detenido o en ejecución); si está ejecutándose indicará los ID de proceso que está utilizando.

```
Se está ejecutando dhcpd (pid 3693)...
```

Tabla 4-7 Mensaje que muestra el estado del servicio dhcpd

Si efectivamente el demonio dhcpd está detenido o con algún problema; se puede probar iniciando o reiniciando el servicio con estos comandos:

```
service dhcpd start
service dhcpd restart
```

En algunas ocasiones cuando se intenta iniciar dhcpd, aparece el siguiente mensaje:

```
st decls to leases file.
Wrote 0 new dynamic host decls to leases file.
Wrote 0 leases to leases file.
Not configured to listen on any interfaces!
```

Tabla 4-8 Mensaje de error de que no están activas las tarjetas de red

Este mensaje de error significa que no existen interfaces ethernet en el servidor o que no están activadas, esto se soluciona con el programa de configuración apropiado; en el caso de Fedora Core existe una herramienta llamada redhat-config-network, con ella se pueden configurar y activar las tarjetas de red del sistema.

Una vez que se haya(n) configurado la(s) tarjeta(s) de red, proporcionando su dirección IP, máscara de red, puertas de enlace, etc., y después de haberla(s) activado; se puede comprobar su estado con el comando ifconfig, el cual mostrará las interfaces ethernet activadas. En este momento, podrá iniciarse el demonio dhcpd sin ningún problema.

Si los comandos start o restart no arrojan errores, significa que dhcpd ahora sí, estará corriendo en el servidor.

4.3.3.2 Verificar la configuración de dhcpd.conf

Como se vio en el capítulo 3, el archivo dhcpd.conf contiene las configuraciones necesarias para que el servicio DHCP funcione correctamente; en ocasiones basta con adaptar ciertas cosas al archivo de ejemplo dhcpd.conf.example; particularmente las direcciones MAC de las terminales. Sin embargo hay situaciones en las que se debe modificar otros parámetros.

En primer lugar se debe verificar que la siguiente línea se encuentre antes del parámetro default-lease-time:

ddns-update-style none;

Esta línea es requerida a partir de la versión 3.0 de DHCP.

Hasta el momento en que la terminal necesita recibir la información para configurar la tarjeta de red y antes de comenzar a bajar el kernel desde el servidor, solo son requeridos ciertos parámetros del archivo dhcpd.conf.

Es necesario especificar las direcciones IP del servidor LTSP, del gateway y del servidor de dominio; en ambientes LTSP generalmente se utiliza un solo servidor que realice estas tres funciones, es decir una misma dirección IP será indicada en estos parámetros.

También se debe verificar que la entrada "hardware ethernet" concuerda exactamente con la dirección MAC de la terminal; ésta dirección se puede conocer cuando se inicia la terminal con la imagen de arranque apropiada.

hardware ethernet 00:E0:06:E8:00:84;

Finalmente se tienen que indicar las direcciones IP que serán asignadas a cada una de las terminales, estas se establecen en el parámetro "fixed-address".

fixed-address 192.168.0.1;

Con esta información bastará para que la terminal pueda configurar la tarjeta de red y comience a bajar la imagen del kernel desde el servidor LTSP.

4.3.3.3 Firewalls

Linux cuenta con herramientas llamadas firewalls, que disminuyen el riesgo de que algún intruso pueda comprometer la seguridad del sistema, sin embargo si estas no son configuradas adecuadamente para funcionar en conjunto con el ambiente LTSP, pueden ocasionar que el servidor no trabaje como se espera.

- Ipchains: Los kernels anteriores al 2.4 tenían la posibilidad de manipular la entrada y salida de los paquetes de datos usando ipchains, el cual utilizaba listas de reglas que se aplicaban a los paquetes en cada paso del proceso de filtrado.

Si en el servidor LTSP, se trabaja con un kernel cuya versión sea menor a la 2.4; se debe teclear el siguiente comando para determinar si ipchains está bloqueando la salida o entrada de paquetes:

```
ipchains -L –v
```

La salida debe ser similar a la siguiente:

```
Chain input (policy ACCEPT: 229714 packets, 115477216 bytes):
Chain forward (policy ACCEPT: 10 packets, 1794 bytes):
Chain output (policy ACCEPT: 188978 packets, 66087385 bytes):
```

Tabla 4-9 Mensaje que muestra si ipchains esta bloqueando los paquetes

En este caso se puede observar que ipchains está aceptando la entrada y salida de los paquetes de datos, por lo que no hay ningún conflicto entre dhcpd e ipchains.

- Iptables: Si la versión del kernel es mayor o igual a la 2.4, se tiene la posibilidad de manipular la entrada y salida de paquetes con ipchains o iptables, pero solo uno a la vez. Generalmente se utiliza iptables debido a que tiene mejoras en el funcionamiento y en el control disponible a la hora de filtrar paquetes.

Se debe ejecutar el siguiente comando para determinar el comportamiento de iptables:

```
iptables -L -v
```

La salida debe ser similar a la siguiente:

```
Chain INPUT (policy ACCEPT 18148 packets, 2623K bytes)
pkts bytes target prot opt in out source destination
Chain FORWARD (policy ACCEPT 0 packets, 0 bytes)
pkts bytes target prot opt in out source destination
Chain OUTPUT (policy ACCEPT 17721 packets, 2732K bytes)
pkts bytes target prot opt in out source destination
```

Tabla 4-10 Mensaje que muestra el comportamiento de iptables

Al igual que en ipchains, si se observa que se está aceptando la entrada y salida de paquetes, no habrá ningún problema entre iptables y dhcpd.

4.3.3.4 Problemas con la terminal

Si después de verificar las configuraciones en el servidor no se logra detectar en él algún problema, es posible que el conflicto se encuentre en la terminal. Para comprobar si la terminal está enviando la petición al servidor se puede observar el archivo /var/log/messages mientras la terminal se inicia.

Se debe teclear el siguiente comando:

```
tail -f /var/log/messages
```

Este comando mostrará las últimas líneas del archivo messages conforme se vayan agregando.

```
server dhcpd: DHCPDISCOVER from 00:50:56:81:00:01 via eth0
server dhcpd: no free leases on subnet WORKSTATIONS
server dhcpd: DHCPDISCOVER from 00:50:56:81:00:01 via eth0
server dhcpd: no free leases on subnet WORKSTATIONS
```

Tabla 4-11 Mensaje para saber si la terminal esta enviando la petición al servidor

Si se muestran algunas líneas en donde aparezca el mensaje "no free leases", entonces dhcpd está corriendo, pero en cambio hay un problema con la terminal debido a que no está haciendo la petición al servidor DHCP.

Esto puede deberse a que la imagen de arranque está defectuosa o la versión no es la correcta; aunque efectivamente dicha imagen corresponda a la tarjeta de red de la terminal, en algunas ocasiones la configuración con la que fue creada provoca que esta no funcione adecuadamente a la hora de arrancar la terminal.

Para solucionar esto, se debe probar iniciando la terminal con distintas versiones de la imagen de arranque; estas versiones se pueden conseguir también en la misma página[15].

En un caso extremo, si ninguna de las imágenes de arranque funciona, se puede probar reemplazando la tarjeta de red de la terminal; ya que puede tener algún problema físico que impida la conectividad con el servidor.

4.3.4 Problemas con el servicio TFTP

La imagen de arranque que permite iniciar las terminales contiene instrucciones o código Etherboot, el cual después de lograr comunicarse con el servidor utiliza el

[15] Página del sitio para descargar la imagen de arranque http://www.rom-o-matic.net.

protocolo TFTP para obtener un kernel Linux a través de la red. Este protocolo es muy simple, pero en ocasiones se generan ciertos problemas cuando se trata de hacerlo funcionar.

En el momento en que la terminal intenta bajar el kernel del servidor, se observa una línea similar a la siguiente:

```
Loading 192.168.0.254:/lts/vmlinuz.tulip .....
```

Tabla 4-12 Mensaje que muestra cuando la terminal intenta bajar el kernel

Posteriormente comienzan a desplegarse una serie de puntos que representan el avance de la descarga; mientras estos puntos continúen apareciendo en la pantalla significa que la imagen del kernel está siendo descargada exitosamente y por lo tanto TFTP está funcionando como se espera.

En el caso de que estos puntos no aparezcan de manera continua entonces quiere decir que hay un problema con el servicio TFTP, las posibles causas pueden ser las siguientes.

4.3.4.1 El demonio tftpd

En Fedora Core, tftp es iniciado por el demonio xinetd; existe un script de inicio llamado /etc/xinetd.d/tftp que contiene la información necesaria para que corra tftpd.

Es posible que el demonio tftpd no esté corriendo, por lo que se puede probar reiniciando el demonio xinetd, el cual reiniciará automáticamente un conjunto de servicios incluyendo tftp. Se debe teclear el siguiente comando:

```
service xinetd restart
```

4.3.4.2 La imagen del kernel no está en el directorio adecuado

La imagen del kernel que será descargado por la terminal, necesita estar en un lugar accesible para el demonio tftpd. Si la opción "-s" está especificada cuando tftpd inicia, entonces cualquier petición de la terminal debe ser relativa al directorio /tftpboot.

Por ejemplo, si el parámetro filename en el archivo de configuración dhcpd.conf es /lts/vmlinuz-2.4.19-ltsp-1, entonces la imagen del kernel deberá estar en la siguiente ruta: /tftpboot/lts/vmlinuz-2.4.19-ltsp-1.

Si no se especifica la opción "-s", entonces en el archivo dhcpd.conf tendrá que especificarse la ruta absoluta de la imagen del kernel, es decir:

```
filename        "/tftpboot/lts/vmlinuz-2.4.19-ltsp-1";
```

Tabla 4-13 Mensaje donde se especifica la ruta absoluta de la imagen del kernel

4.3.5 Problemas con el servicio NFS

NFS permite montar un sistema de archivos remoto a cada una de las terminales, éste sistema de archivos será realmente un directorio compartido que será exportado por el servidor; por lo que éste se debe configurar adecuadamente para que permita el acceso de las terminales.

Existen varios factores que pueden impedir que un sistema de archivos raíz pueda ser montado en las terminales:

4.3.5.1 El proceso Init no puede ser iniciado

Cuando la terminal intenta montar el sistema de archivos raíz, puede mostrar el siguiente error:

```
Kernel panic: No init found. Try passing init= option to kernel.
```

Tabla 4-14 Mensaje de error al intentar montar el sistema de archivos raíz

Este mensaje indica que el contenido del directorio exportado por el servidor no es el adecuado, o no contiene la información que necesita el kernel para continuar con el proceso de configuración del sistema.

Los scripts de configuración del LTSP pudieron haber tenido errores en la creación del directorio a exportar, por lo que es probable que el directorio /opt/ltsp/i386 esté vacío.

Se puede listar el contenido de este directorio para verificar que tenga la estructura correcta y la cual es la que espera encontrar el kernel de la terminal.

```
ls -l  /opt/ltsp/i386
```

Los siguientes directorios deberán estar presentes:

bin dev etc lib mnt oldroot opt proc root sbin tmp usr var

4.3.5.2 El servidor retorna el error -13

Si se muestra en la terminal el siguiente error:

```
Root-NFS: Server returned error -13 while mounting /opt/ltsp/i386
```

Tabla 4-15 Mensaje de error que muestra que el directorio i386 no esta en exports

Esto indicará que el directorio /opt/ltsp/i386 no está especificado en el archivo de configuración /etc/exports.

Es posible observar el contenido del archivo /var/log/messages para determinar alguna posible causa.

```
Jul 20 00:28:39 jamlap rpc.mountd: refused mount request from ws001
for /opt/ltsp/i386 (/): no export entry
```

Tabla 4-16 Muestra el contenido del archivo /var/log/messages

Este mensaje se puede interpretar fácilmente y confirma que efectivamente el directorio que la terminal pretende montar, no está siendo exportado por el servidor.

Para solucionar esto, basta con verificar el contenido del archivo /etc/exports y añadir si es que no existe, la siguiente línea:

```
/opt/ltsp/i386     192.168.0.0/255.255.255.0(ro,no_root_squash,sync)
```

Tabla 4-17 Muestra el contenido del archivo /

4.3.6 Problemas con el servidor X

La implementación del LTSP puede llegar a ser complicada debido a la gran diversidad de hardware, y un claro ejemplo de esto es la correcta configuración del servidor X.

Si se está utilizando una tarjeta de video y un monitor relativamente modernos, es casi seguro que estén soportados por el servidor XFree86 y por lo tanto no habrá mucho problema en que funcionen adecuadamente; sin embargo cuando se trata de equipos antiguos es necesario realizar algunos ajustes adicionales al servidor.

Los scripts de inicialización del LTSP tratarán de ejecutar el servidor X en las terminales, para que éste se comunique con el Display Manager del servidor. Si por alguna razón el servidor X no puede ser iniciado al primer intento, se probará hacerlo en 10 ocasiones más hasta que finalmente se muestre un mensaje de error si no se tuvo éxito en la conexión.

Cuando sucede esto, lo primero que se debe intentar es cambiar el servidor X especificado en el archivo lts.conf; por default se tendrá configurado el servidor XFree86 pero si la tarjeta de video es algo antigua es muy probable que se deba usar otro servidor X, por ejemplo Xvesa.

Antes de poder modificar el archivo lts.conf, será necesario instalar el paquete correspondiente al servidor Xvesa; el cual puede obtenerse en la página oficial del LTSP.

Para instalar el paquete se tiene que teclear el siguiente comando:

```
rpm –ivh  ltsp_x_vesa-3.0-1.k12ltsp.2.3.0.i386.rpm
```

Este paquete instalará los componentes necesarios para que el servidor Xvesa esté disponible para las terminales, para ello se creará el archivo Xvesa en la ruta:

```
/opt/ltsp/i386/usr/X11R6/bin/
```

Tabla 4-18 Muestra la ruta del archivo Xvesa

Una vez instalado el paquete, se requiere modificar el parámetro XSERVER en la sección de la terminal con problemas; se tendrá que cambiar el valor Xfree86 por Xvesa.

```
XSERVER = Xvesa
```

Después de efectuar los cambios correspondientes, se debe reiniciar la terminal para verificar si el servidor Xvesa soporta su tarjeta de video y monitor.

En caso de que el servidor Xvesa tampoco funcione, se tendrá que recurrir a otras alternativas, por ejemplo los servidores vga o svga; por lo cual es necesario repetir el proceso que se utilizó con Xvesa.

Hay ocasiones en que el servidor X si soporta a la tarjeta de video de la terminal, sin embargo la configuración del monitor no es la apropiada.

En modelos de monitores recientes, la configuración automática es suficiente, pero en monitores antiguos es necesario especificar algunos parámetros adicionales; los cuales definen la resolución así como la frecuencia de refresco de la pantalla.

Se tendrá que agregar los siguientes parámetros al archivo lts.conf, en la sección de la terminal:

```
X_HORZSYNC          = "30.9-49.0"
X_VERTREFRESH  = "50.0-100.0"
X_MODE_0          = 800x600
```

O bien se podrá especificar un modeline en el parámetro X_MODE_0, por ejemplo:

```
X_HORZSYNC  = "30.9-49.0"
X_VERTREFRESH = "50.0-100.0"
X_MODE_0 = 800x600  40     800  840  968 1056   600 601 605 628
    +hsync +vsync
```

Con las modificaciones antes descritas es muy probable que el servidor X inicie correctamente en cada una de las terminales, pero aun es necesario que éste logre establecer una conexión con el Display Manager que corre en el servidor.

4.3.7 Problemas con el Display Manager

El Display Manager (Administrador de Sesiones Gráficas) es un demonio que corre en el servidor, el cual espera que un servidor X se ponga en contacto con él. En el momento en que se establece la conexión se mostrará una caja de diálogo de inicio de sesión, para que el usuario pueda realmente acceder a las aplicaciones del servidor.

Los tres Display Managers más comunes en Linux son:

XDM - Forma parte del sistema estándar de X Window.
GDM - El "Gnome Display Manager" forma parte del entorno de escritorio Gnome.
KDM - El "KDE Display Manager" forma parte del entorno de escritorio KDE.

En ocasiones el servidor X logra iniciar correctamente en la terminal, pero no puede establecer una conexión con el Display Manager; cuando sucede esto, se presenta una pantalla gris con un cursor en forma de X en lugar de la caja de diálogo de inicio de sesión. Las posibles razones son las siguientes:

4.3.7.1 El Display Manager no está en ejecución

Para determinar si algún Display Manager está corriendo en el servidor se puede utilizar el siguiente comando:

```
netstat -ap | grep xdmcp
```

La salida de éste comando mostrará si hay algún proceso escuchando en el puerto xdmcp (177).

```
udp 0 0 *:xdmcp *:* 1255/gdm
```

Tabla 4-19 Muestra la salida para determinar si algún display manager esta activo

En este ejemplo se puede observar que gdm está corriendo con el PID 1255 y está escuchando en el puerto de XDMCP. En caso de que no se muestre salida alguna, significará que ningún Display Manager está corriendo; por lo cual será necesario reiniciar el sistema para reactivar el Display Manager por defecto que utiliza el servidor.

4.3.7.2 El Display Manager ignora peticiones de máquinas remotas

Si se ha comprobado que algún Display Manager está corriendo, entonces es posible que haya sido configurado para ignorar peticiones XDMCP desde máquinas remotas; para solucionar esto, es necesario verificar los archivos de configuración correspondientes al Display Manager utilizado.

4.3.7.2.1 XDM

En el archivo xdm-config la siguiente línea deberá estar comentada para que se admitan peticiones remotas:

```
#    DisplayManager.requestPort: 0
```

Y el archivo Xaccess deberá contener una línea que comience con un asterisco (*):

```
*    # Aceptar peticiones remotas
```

Ambos archivos generalmente se encuentran en la ruta /etc/X11/xdm/ pero si no es así, se pueden buscar con el comando locate.

4.3.7.2.2 KDM

El archivo kdmrc contiene una sección con la etiqueta [xdmcp], dentro de la cual hay un parámetro que deberá estar configurado de la siguiente manera:

```
Enable = true    #true habilita las peticiones remotas
```

Este archivo generalmente se encuentra en la ruta /etc/kde/kdm/ pero si no es así, se puede buscar con el comando locate.

4.3.7.2.3 GDM

El archivo gdm.conf contiene una sección con la etiqueta [xdmcp], dentro de la cual hay un parámetro que deberá estar configurado de la siguiente manera:

```
Enable = true    #true habilita las peticiones remotas
```

Este archivo generalmente se encuentra en la ruta /etc/X11/gdm/ pero si no es así, se puede buscar con el comando locate.

4.3.7.3 Conflicto del Display Manager con el DNS

Si el Display Manager definitivamente está en ejecución y está escuchando las peticiones de máquinas remotas, entonces el problema puede deberse a un simple error en el mapeo de direcciones IP a nombres, por lo que el Display Manager no sabrá con que terminal establecer la comunicación.

Para solucionar este problema es necesario configurar el archivo hosts, agregando una entrada para cada una de las terminales, por ejemplo:

```
192.168.0.1   ws001    ws001.localdomain   ws001
192.168.0.2   ws002    ws002.localdomain   ws002
```

4.3.8 Archivos de sistema

Finalmente, después de resolver los problemas detectados y realizar las modificaciones correspondientes a los archivos de configuración estos quedarán así:

```
# Archivo de configuración dhcpd.conf
ddns-update-style            none;
default-lease-time           21600;
max-lease-time               21600;
option subnet-mask           255.255.255.0;
option broadcast-address     192.168.0.255;
option routers               192.168.0.254;
option domain-name-servers   192.168.0.254;
option domain-name           "yourdomain.com";
option root-path             "192.168.0.254:/opt/ltsp/i386";
option option-128 code 128 = string;
option option-129 code 129 = text;

shared-network WORKSTATIONS {
   subnet 192.168.0.0 netmask 255.255.255.0 {
   }
}

group    {
   use-host-decl-names      on;
   option log-servers       192.168.0.254;
   host ws001 {
      hardware ethernet     00:60:8C:CA:BC:FF;
      fixed-address         192.168.0.1;
      filename              "/lts/vmlinuz-2.4.19-ltsp-1";
      option option-128     e4:45:74:68:00:00;
      option option-129     "NIC=3c509";
   }
   host ws002 {
      hardware ethernet     00:A0:24:24:F6:0F;
      fixed-address         192.168.0.2;
      filename              "/lts/vmlinuz-2.4.19-ltsp-1";
      option option-128     e4:45:74:68:00:00;
      option option-129     "NIC=3c509";
   }
   host ws003 {
      hardware ethernet     00:80:AD:3B:8F:E4;
      fixed-address         192.168.0.3;
      filename              "/lts/vmlinuz-2.4.19-ltsp-1";
   }
}
```

Tabla 4-20 Muestra le archivo completo de configuración exports

```
# Archivo de configuración exports

/opt/ltsp/i386                  192.168.0.0/255.255.255.0(ro,no_root_squash,sync)
/var/opt/ltsp/swapfiles   192.168.0.0/255.255.255.0(rw,no_root_squash,async)
```

Tabla 4-21 Muestra el archivo completo de configuración exports

```
# Archivo de configuración hosts.allow

bootpd:   0.0.0.0
in.tftpd:  192.168.0.
portmap:   192.168.0.
mountd:    192.168.0.
ALL:        127.0.0.1
ALL:      192.168.0.254
ALL:      192.168.0.1
ALL:      192.168.0.2
ALL:      192.168.0.3
```

Tabla 4-22 Muestra el archivo completo de configuración hosts.sllow

```
# Archivo de configuración lts.conf

[Default]
      SERVER          = 192.168.0.254
      XSERVER         = auto
      X_MOUSE_PROTOCOL  = "PS/2"
      X_MOUSE_DEVICE    = "/dev/psaux"
      X_MOUSE_BUTTONS   = 3
      X_USBMOUSE_PROTOCOL= "IMPS/2"
      X_USBMOUSE_DEVICE  = "/dev/input/mice"
      X_USBMOUSE_BUTTONS = 5

      XkbSymboles      = "us(pc101)"
      XkbModel         = "pc101"
      XkbLayout        = "us"

      USE_XFS          = N
      LOCAL_APPS       = N
      RUNLEVEL         = 2

      SOUND            = Y

      VOLUME           = 75

[ws001]
      XSERVER          = Xvesa
      LOCAL_APPS       = N
      USE_NFS_SWAP     = N
      SWAPFILE_SIZE    = 48m
      RUNLEVEL         = 5
      X_HORZSYNC       = "30.9-49.0"
      X_VERTREFRESH    = "50.0-100.0"
      X_MODE_0         = 800x600
[ws002]
      XSERVER          = Xvesa
      LOCAL_APPS       = N
```

```
    USE_NFS_SWAP      = N
    SWAPFILE_SIZE     = 48m
    RUNLEVEL          = 5
    X_HORZSYNC        = "30.9-49.0"
    X_VERTREFRESH     = "50.0-100.0"
    X_MODE_0          = 800x600
[ws003]
    XSERVER           = Xvesa
    LOCAL_APPS        = N
    USE_NFS_SWAP      = N
    SWAPFILE_SIZE     = 48m
    RUNLEVEL          = 5
    X_MODE_0          = 800x600
```

Tabla 4-23 Muestra el archivo completo de configuración lts.conf

4.4 Conexión en red

De acuerdo con la metodología propuesta, cuando se logra dar solución a los problemas detectados y todas las terminales inician una sesión gráfica individualmente; el siguiente paso es probar su funcionamiento conectándolas en red.

Este proceso es relativamente sencillo, pero hay que considerar algunos aspectos importantes en su implementación, los cuales básicamente son:

- Uso de cable UTP directo en lugar de cable cruzado
- Uso de un switch en lugar de un hub para garantizar un mejor rendimiento en la red
- Asignación correcta de direcciones IP para evitar conflictos entre las terminales

Tomando en cuenta los puntos anteriores, los pasos para implementar la red son los siguientes:

1. Conectar el servidor al primer puerto del switch mediante un cable directo
2. Conectar la terminal 1 al segundo puerto del switch, la segunda terminal al tercer puerto y así sucesivamente, todas estas conexiones mediante cable directo
3. Encender el servidor e iniciar los servicios necesarios del LTSP
4. Encender el switch
5. Encender cada una de las terminales y autenticarse para iniciar una sesión remota en el servidor

El siguiente diagrama muestra la configuración de los equipos al estar conectados en red:

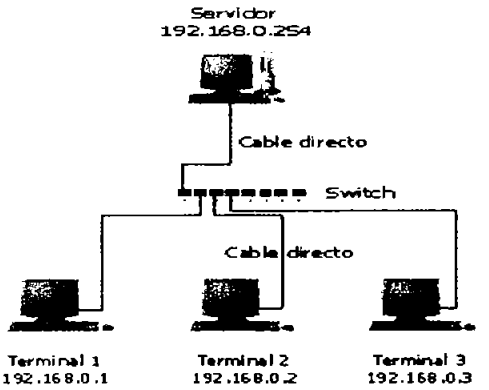


Figura 4-2 Diagrama que muestra la configuración de los equipos en red

La siguiente figura muestra a las terminales con la pantalla de bienvenida del LTSP al estar en red.

Figura 4-3 Imagen de las terminales en red con la pantalla de bienvenida del LTSP

4.5 Conexión a internet

Inicialmente, la red de terminales gráficas ha sido configurada para funcionar de manera local, esto de por si ya ofrece muchas ventajas, las cuales han sido mencionadas a lo largo de este trabajo, pero debido a que actualmente el acceso a Internet es indispensable, se optó por configurar la red local para que tenga acceso a éste.

Para establecer una conexión a Internet, debe existir un medio por el cual la red local se pueda comunicar con las redes externas; esto básicamente se logra con una tarjeta de red adicional en el servidor, lo que permite que éste también tenga la función de un gateway. Aunque es posible utilizar una sola tarjeta de red creando un "alias", esto no es recomendable debido al tráfico y a la carga que se generaría en dicho dispositivo.

Al configurar el servidor para que tenga salida a Internet por medio de la segunda tarjeta de red (eth1), automáticamente todas las terminales también tendrán acceso. Es necesario entender que en una red implementada con el LTSP, cualquier usuario conectado en un ambiente grafico, es en realidad un usuario local del servidor, por lo tanto, cualquier aplicación que lo requiera puede usar la misma conexión a Internet del servidor sin ningún inconveniente.

El programa redhat-config-network permite configurar de forma sencilla los dispositivos de red instalados en el sistema, éste puede ser ejecutado desde una terminal con el siguiente comando:

```
redhat-config-network &
```

Este programa detecta de forma automática los dispositivos de red instalados y los muestra en la sección "Dispositivos", en este caso deben aparecer dos dispositivos: eth0 y eth1.

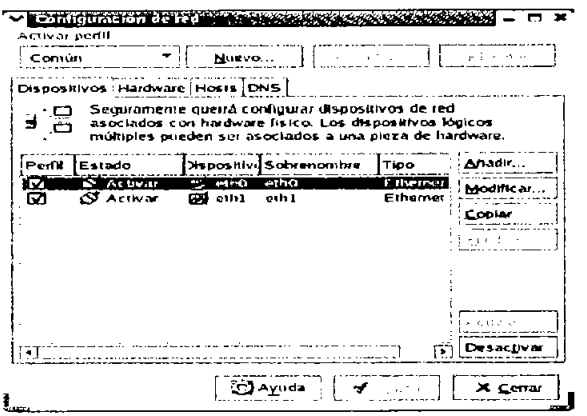


Figura 4-4Pantalla donde se muestran los dispositivos lógicos de red

En la pestaña Hardware, se debe verificar que se haya detectado correctamente la marca y modelo de las tarjetas de red; esto es necesario debido a que pueden ocurrir errores en los controladores, si estos no son los apropiados.

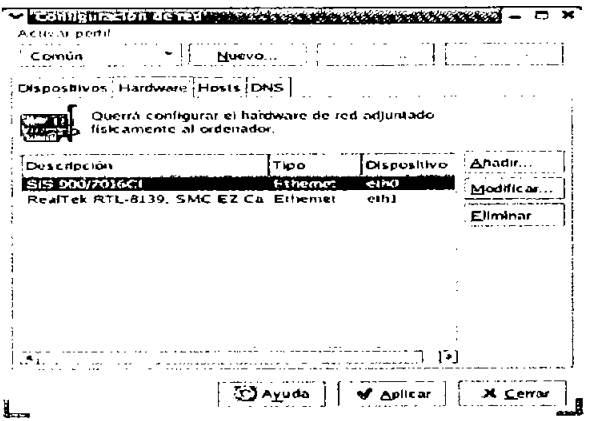


Figura 4-5 Pantalla donde se muestran los dispositivos físicos de red

La tarjeta eth0 permitirá comunicar al servidor con la red de terminales gráficas, para configurarla se debe regresar a la pestaña "Dispositivos" y seleccionar el dispositivo eth0; hacer click en "Modificar".

En la siguiente pantalla, se tiene que desactivar la opción "Obtener las configuraciones de direcciones IP automáticamente con dhcp", se activará la opción "Configurar las direcciones IP de manera estática". Se debe ingresar la dirección IP 192.168.0.254, la máscara de subred 255.255.255.0 y la puerta de enlace 192.168.0.254. Hacer click en "OK" para continuar.

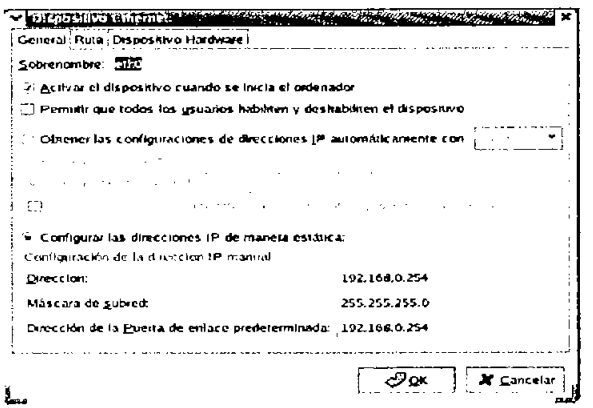


Figura 4-6 Configuración del dispositivo eth0

La tarjeta eth1 permite comunicar al servidor con las redes externas, en realidad es en este dispositivo donde se configura el acceso a Internet, por lo que se debe seleccionar y hacer click en "Modificar". En la siguiente pantalla, se tiene que desactivar la opción "Obtener las configuraciones de direcciones IP automáticamente con dhcp", se activará la opción "Configurar las direcciones IP de manera estática". Se debe ingresar la dirección IP 132.248.173.84, la máscara de subred 255.255.255.0 y la puerta de enlace 132.248.173.254. Hacer click en "OK" para continuar.

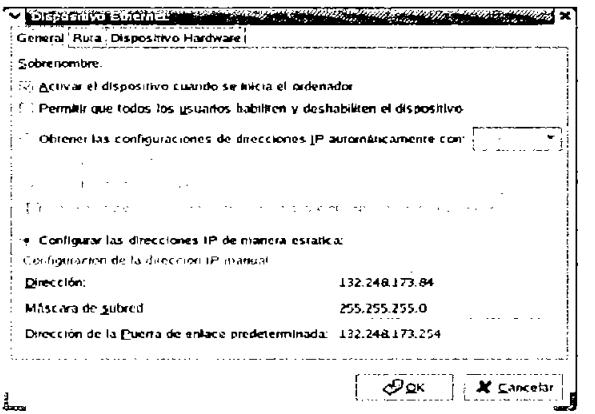


Figura 4-7 Configuración del dispositivo eth1

En la pestaña "DNS", se debe ingresar el nombre de host "Electromartin", el DNS primario 132.248.204.1 y el DNS secundario 132.248.10.2. En la sección "Ruta de búsqueda DNS", se debe añadir el dominio "aragon.unam.mx". Una vez ingresados los datos, hacer click en "Aplicar" para confirmar los cambios.

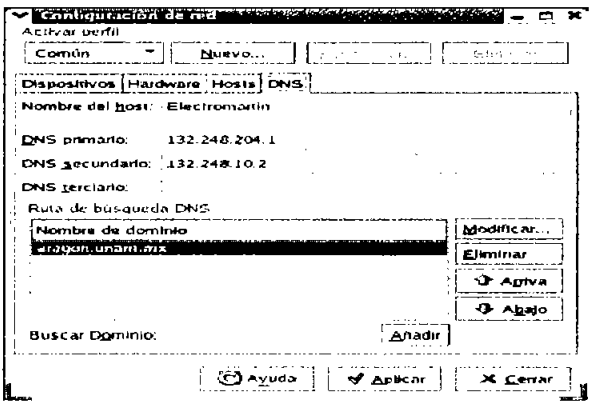


Figura 4-8 Configuración del DNS

4.6 Software específico

El LTSP es un claro ejemplo de las ventajas y posibilidades que el software libre puede ofrecer, ya que mediante la adecuada combinación de diferentes servicios y programas se logró implementar la red de terminales gráficas en el laboratorio de electrónica.

Como ya se había mencionado, dicha red proporciona entre otras cosas, un sistema centralizado de almacenamiento de información y la posibilidad de utilizar una gran variedad de aplicaciones, la mayoría de ellas distribuidas bajo la licencia GNU/GPL.

Sin embargo, no es conveniente instalar de una forma indiscriminada programas que tal vez nunca serán utilizados; para evitar esto se pensó en la necesidad de los alumnos de contar con software relacionado con la electrónica, el cual puede clasificarse en tres principales grupos: diseño, simulación y programación de circuitos.

Actualmente ya se cuenta con software de este tipo, por ejemplo: el Electronics Workbench, Pspice, Warp, Mplab, entre otros; pero debido a que estos programas corren en Windows se optó por buscar software propio de Linux que pudiera ser útil en el laboratorio y que tuviera características semejantes a los anteriores.

Cabe mencionar que el software para Windows podrá seguir utilizándose debido a que las terminales han sido configuradas para que puedan seguir arrancando este sistema operativo si así se requiere. En Internet hay muchos programas de electrónica disponibles, pero solo se describirán los que se consideraron más funcionales y amigables para el usuario.

4.6.1 Diseño

Este tipo de software es utilizado para crear diagramas o esquemas que representan gráficamente un circuito electrónico, éstos pueden imprimirse o exportarse a otros programas mediante la generación de un archivo compatible. Los programas de diseño no tienen la capacidad de simular o interpretar el funcionamiento del diagrama, sin embargo son muy útiles debido a que permiten generar fácilmente la estructura de casi cualquier circuito eléctrico - electrónico.

4.6.1.1 Eagle

Es un software de diseño de esquemas PCB (Printed Circuit Boards) con motor de autotrazado y una gran variedad de componentes. Es comercial, pero existe una licencia freeware (educacional) que limita el tamaño máximo de la placa de circuito impreso.

El programa consta de tres módulos principales:

- Editor de circuito impreso (Layout Editor)
- Editor de diagramas esquemáticos (Schematic Editor)
- Organizador de pistas (Autorouter)

Estos tres módulos están integrados en una sola interfaz de usuario, lo que permite crear fácilmente el diseño para la placa, a partir del circuito implementado en el editor esquemático.

La versión de evaluación del software permite crear circuitos impresos de hasta dos caras y con área máxima de 100 x 80 mm, suficiente para una gran variedad de circuitos sencillos.

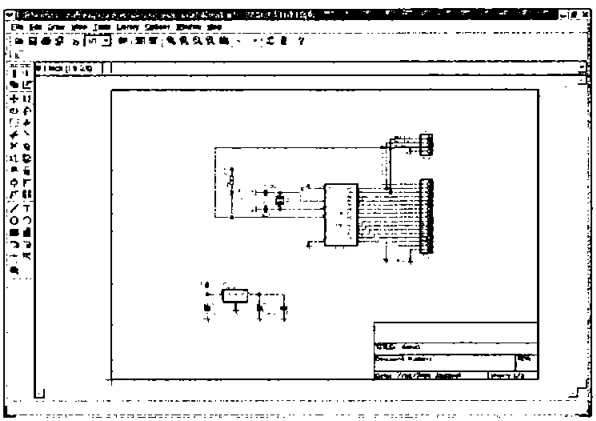


Figura 4-11 Imagen del programa Eagle

Para la instalación de Eagle, se debe descargar la última versión del código fuente[16] y seguir los pasos descritos en el capítulo 2 (métodos de instalación).

4.6.1.2 Xcircuit

Xcircuit es un programa que permite dibujar de una manera sencilla, diagramas de circuitos eléctricos de gran calidad, los cuales pueden ser exportados a otros programas como el Spice. También es posible realizar esquemas de diversas temáticas, debido a que pueden utilizarse bibliotecas de distintos objetos.

[16] Página para descargar el código http://www.cadsoft.de/info.htm

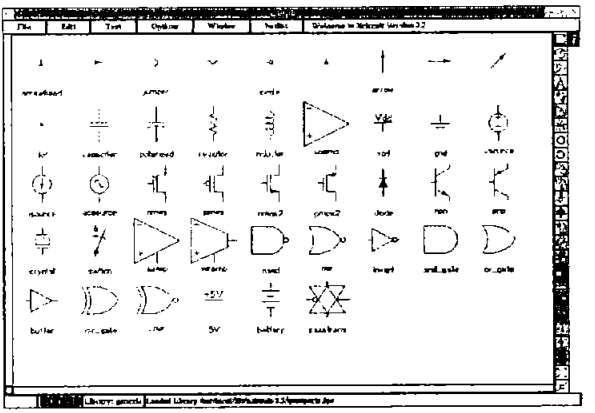


Figura 4-12 Imagen del programa Xcircuit

Para la instalación de Xcircuit, se debe descargar la última versión del código fuente[17] y seguir los pasos descritos en el capítulo 2 (métodos de instalación).

4.6.2 Simulación

Estos programas también permiten crear diagramas de circuitos electrónicos, pero tienen la gran ventaja de que se puede analizar el comportamiento físico del circuito; se puede determinar el voltaje, corriente o los estados lógicos por los que va pasando cada componente. Por lo general, los simuladores cuentan con varias herramientas entre las cuales se encuentran: los diagramas de componentes, diagramas de onda y cálculo de parámetros.

4.6.2.1 Klogic

Klogic es una herramienta para construir y simular circuitos digitales que contienen componentes estándar como AND, OR, XOR, Flip Flops RS y JK. Si se requiere construir circuitos más complejos y reutilizables se tiene la opción de diseñar subcircuitos.

Mientras se está construyendo el circuito, la simulación está ejecutándose de forma permanente, además para pruebas extendidas el programa tiene la capacidad de simular el circuito paso a paso para reconocer los estados por los que éste va pasando. También es posible desplegar la señal de flujo de los componentes de un circuito de forma gráfica e intuitiva.

[17] Página para descargar el código http://bach.ece.jhu.edu/~tim/programs/xcircuit/welcome.html

paquetes; lo que permite que su uso sea fácilmente extendido para nuevas áreas de aplicación.

Los principales componentes de Ksimus son:

- Compuertas AND, NAND, OR, NOR, XOR, XNOR
- Flip Flops RS, JK y JK-MS
- Generador de pulsos
- LED
- Latches D
- Contador
- Display de 7 segmentos
- Multiplexor y demultiplexor
- Generador de ondas
- Aritméticos (add, sub, mul, div, mod, abs)
- Trigonométricos (sin, asin, cos, acos, tan, atan, atan2)
- Hiperbólicos (sinh, asinh, cosh, acosh, tanh, atanh)
- Exponenciales (log, log10, exp, pow, square)

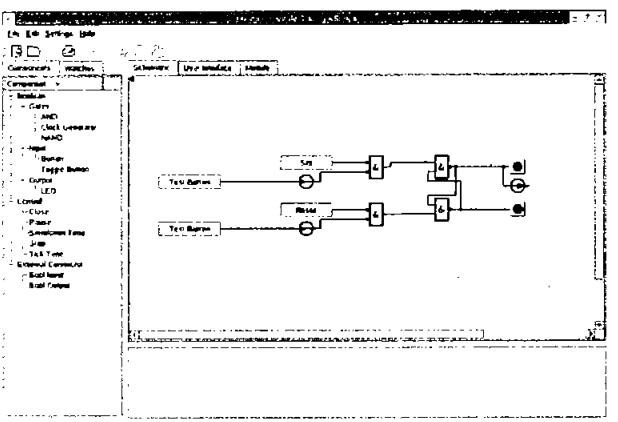


Figura 4-15 Imagen del programa Ksimus

Para la instalación de Ksimus, se debe descargar la última versión del código fuente[19] y seguir los pasos descritos en el capítulo 2 (métodos de instalación).

[19] Página para bajar el código fuente http://ksimus.berlios.de/index.html

4.6.2.3 Tkgate

Tkgate es un simulador de circuitos digitales, basado en un editor gráfico tcl/tk. Soporta una amplia gama de componentes básicos, pero además se tiene la capacidad de construir módulos mediante el diseño jerárquico de circuitos.

Los principales componentes de Tkgate son:

- Compuertas AND, NAND, OR, NOR, XOR, XNOR
- Transistores NMOS y PMOS
- Buffers triestado
- Componentes ALU (sumadores y multiplicadores)
- Elementos de memoria (registros, RAMs, ROMs)

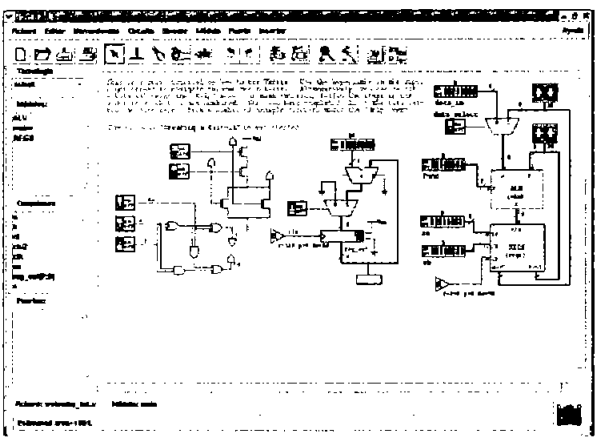


Figura 4-16 Imagen del programa Tkgate

Para la instalación de Tkgate, se debe descargar la última versión del código fuente[20] y seguir los pasos descritos en el capítulo 2 (métodos de instalación).

4.6.2.4 Vipec

Vipec es una poderosa herramienta para el análisis de circuitos de alta frecuencia y redes eléctricas lineales. Es adecuado para simular en el dominio de la frecuencia, circuitos relacionados con las comunicaciones, como: amplificadores, filtros y líneas de transmisión.

[20] Página para bajar el código fuente http://www.tkgate.org

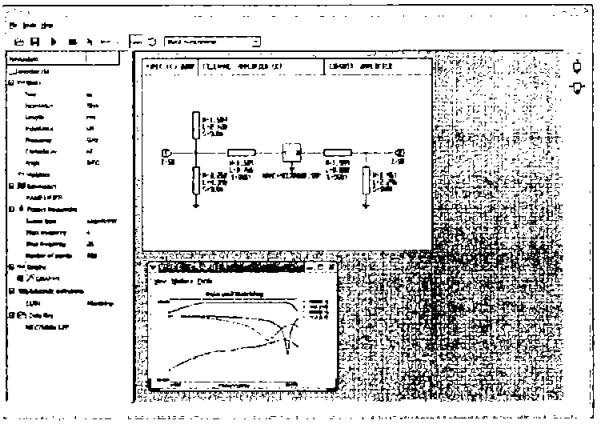


Figura 4-17 Imagen del programa Vipec

Para la instalación de Vipec, se debe descargar la última versión del código fuente[21] y seguir los pasos descritos en el capítulo 2 (métodos de instalación).

4.6.2.5 Gpsim

Este entorno de simulación para PIC's soporta microprocesadores de 12, 14 y 16 bits, además tiene la capacidad de simular en tiempo real sistemas con un ciclo de reloj de 20 MHz.

GPSIM cuenta con un depurador, un desensamblador, un manejador de estímulos, un mapa de memoria y algunos plugins que permiten simular componentes externos al PIC como una pantalla LCD.

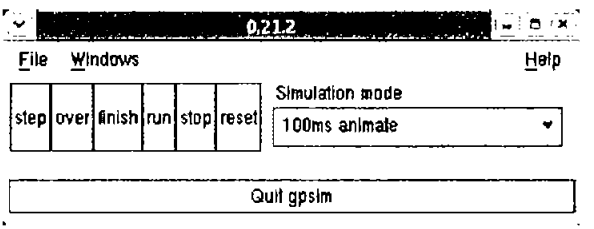


Figura 4-18 Imagen del programa Gpsim

[21] Página para bajar el código fuente http://vipec.sourceforge.net

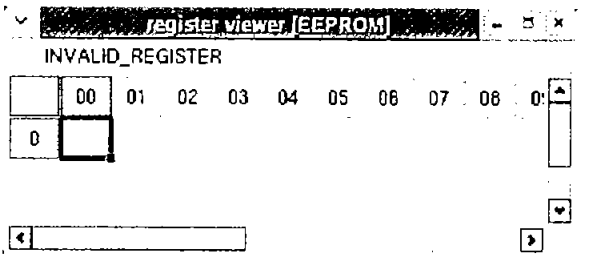


Figura 4-19 Imagen de un ejemplo de programación del programa Gpsim

Para la instalación de Gpsim, se debe descargar la última versión del código fuente[22] y seguir los pasos descritos en el capítulo 2 (métodos de instalación).

4.6.3 Programación

Estas aplicaciones permiten diseñar mediante lenguajes de alto nivel, la estructura lógica de circuitos programables, por ejemplo: memorias, microcontroladores, microprocesadores, PAL, GAL, etc.

Una vez que el diseño es terminado, éste tiene que ser copiado en el chip correspondiente, lo cual es posible utilizando un dispositivo especial denominado "grabador". En ocasiones es posible simular el comportamiento del circuito antes de ser grabado, lo que permite la optimización y depuración del diseño.

4.6.3.1 BlueHDL

BlueHDL es una suite de herramientas VHDL, que consiste en un editor y compilador VHDL, un motor de simulación, y una interfaz de visualización de ondas.

Permite diseñar con lenguajes de alto nivel logrando desarrollar circuitos programables, es una buena opción ya que para todo circuito por sencillo que sea se busca realizarlo programando un microcontrolador.

[22] Página para bajar el código fuente http://www.dattalo.com/gnupic/gpsim.html

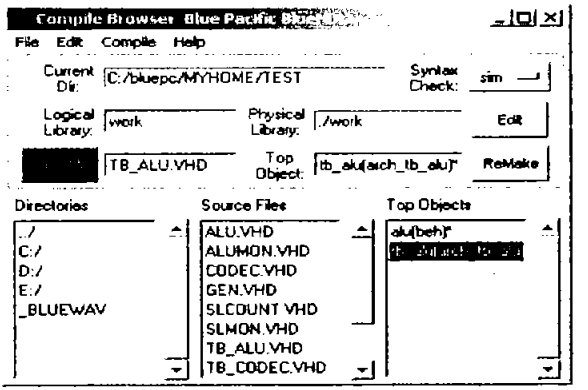


Figura 4-20 Imagen del programa BlueHDL

Para la instalación de BlueHDL, se debe descargar la última versión del código fuente[23] y seguir los pasos descritos en el capítulo 2 (métodos de instalación).

Con todas las instalaciones y configuraciones previas se comprobó que efectivamente cada una de las terminales inició una sesión gráfica adecuadamente, apoyándonos en la parte de detección y solución de problemas; además la red y la conexión a internet funcionó tal y como se esperaba. Ahora analizaremos el rendimiento del servidor para conocer cuántas terminales realmente pueden funcionar adecuadamente.

[23] Página para bajar el código fuente http://www.bluepc.com/index.html

PRUEBAS Y ANÁLISIS DE LOS RESULTADOS

OBJETIVO ESPECÍFICO

En primer lugar se comprobará el rendimiento del servidor con sus tres terminales ejecutando diversas aplicaciones, también se analizarán algunas medidas y acciones que garanticen un mínimo de seguridad y funcionalidad en la red; y por último se darán las pautas para retomar posibles proyectos relacionados o simplemente para mejorar los servicios que proporciona el servidor.

5

INTRODUCCIÓN

Para el desarrollo de este apartado se planteó la necesidad de analizar el rendimiento del servidor para tres terminales, la idea básica es dar a conocer el porcentaje de recursos que requieren las terminales para funcionar apropiadamente sin colapsarse o alentar al servidor. Los valores se obtuvieron con la ayuda del comando TOP y en el caso del procesador sus valores son dados en porcentajes para una mejor comprensión.

Por otro lado se puntualizan las ventajas de utilizar el sistema operativo Linux que proporciona algunas características con respecto a la administración y seguridad del sistema, no es necesario realizar configuraciones complejas para garantizar la seguridad que se desea, ya que el propio sistema viene con valores predefinidos que proporcionan un nivel aceptable de seguridad; pero no está de más tomar ciertas medidas y precauciones las cuales se darán en dicho apartado.

Con respecto a la administración, Linux es un sistema realmente amigable y potente, facilita el manejo de usuarios y lo más importante es que mantiene espacios restringidos para un mejor control del sistema. Es importante mencionar que la administración y la seguridad están íntimamente ligados ya que sino hubiera un control estricto del espacio de usuarios, ésto podría ser una causa para que se realizara un ataque al sistema comprometiendo la seguridad del mismo.

Una vez que se ha comprobado que el sistema es realmente estable y se tiene un control de los archivos de sistema, así como de los recursos que se le brindan a los usuarios para no permitirles un uso inapropiado de los mismos; se puede garantizar que se cuenta con una red estable, fácil de administrar y con un nivel aceptable de seguridad.

Pero eso no es todo, ya que es posible hacer mejoras o innovaciones; hay varias aplicaciones u opciones que se pueden implementar para mejorar los servicios brindados en la red de terminales gráficas.

5.1 Pruebas de rendimiento

En este apartado se probará el rendimiento del servidor y para esto se tendrá que encender terminal por terminal para tener la oportunidad de ir monitoreando las peticiones de recursos que se hacen al servidor, y a su vez tener una relación de la cantidad de memoria que se requiere para soportar n cantidad de terminales funcionando adecuadamente.

Estas pruebas ayudarán a comprobar que Linux administra eficientemente los recursos disponibles, principalmente de memoria y de procesamiento del sistema, los valores de las tablas muestran claramente cuánto se consume de los recursos del servidor pero a la vez es posible visualizar que se llega a un estado de estabilidad que impide que el sistema deje de funcionar.

El servidor cuenta con las siguientes características:

Procesador	700 Mhz
Memoria	376640 Kb
Swap	786432 Kb

Se recurrió al siguiente comando para obtener la información del uso de procesador, memoria y swap del servidor:

```
top
```

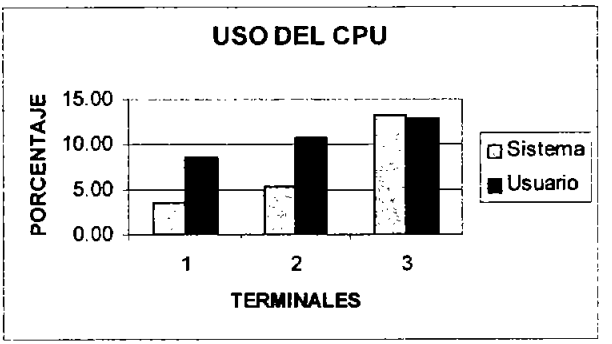
Los resultados obtenidos al momento de encender las tres terminales, se muestran en la siguiente tabla:

TERMINAL	CPU (%)		MEM (Kb)	SWAP (Kb)
	SISTEMA	USUARIO		
1	3.50	8.50	348868	0
2	5.30	10.80	364496	0
3	13.30	12.90	371024	0

Tabla 5-1 Resultados obtenidos al momento de encender las tres terminales

En la siguiente gráfica se puede notar que el procesador no tiene gran desgaste al arrancar las terminales y esto sin tomar en cuenta lo que requiere el servidor. Se puede ver que con la primer terminal se utilizó un 8.5% entonces con una segunda debería de consumir un 17% y a su vez con una tercera gastaría un 25.5% de la capacidad del procesador.

Pero esto no es así ya que realmente sólo se empleo un 13% aproximadamente y no el 25% que se tenía estimado, este comportamiento se debe principalmente a que el kernel de Linux administra eficientemente el uso del procesador; el planificador de procesos es el encargado de gestionar el uso de este recurso ya que verifica la prioridad de cada uno de los procesos y los manda a ejecución solo cuando es necesario, optimizando en gran medida los tiempos del procesador.



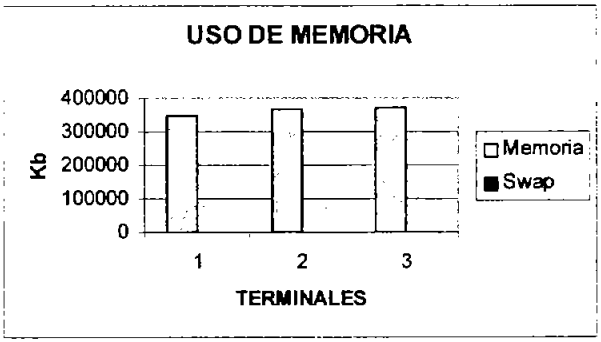
Gráfica 5-1 El uso del CPU

Se puede observar también que la memoria se está utilizando prácticamente en su totalidad, a pesar de que solo es el arranque y aún no se ha ejecutado alguna aplicación en las terminales; cabe mencionar que el sistema aun no ha requerido el uso de la memoria swap.

Bajo estas circunstancias se podría pensar que el sistema esta desperdiciando la memoria, pero no es así, ya que Linux utiliza toda la memoria libre para caché de una forma más dinámica. En un estado estable prácticamente toda la memoria principal está ocupada con fragmentos de procesos, por lo que se tendrá acceso directo a la mayor cantidad de procesos posibles disminuyendo el tiempo en la ejecución de un programa.

El kernel de Linux soporta ejecutables cargados por demanda de páginas, donde solo aquellos segmentos de un programa que se requieren son leídos del disco y puestos en la memoria.

Así, si varias instancias de un programa son ejecutadas simultáneamente, sólo una copia del código del programa estará en memoria y por esta razón el sistema se mantiene estable al ejecutar los mismos programas en las terminales.



Gráfica 5-2 El uso de la memoria

Una vez que las terminales han iniciado una sesión gráfica, es necesario verificar el rendimiento del servidor pero en este caso corriendo aplicaciones del usuario. Para esto, se eligieron 3 aplicaciones que se consideraron como las más utilizadas por los alumnos (Navegador Web, procesador de textos y hoja de cálculo), así como 3 programas específicos para el diseño y simulación de circuitos electrónicos (Xcircuit, Tkgate y Bluehdl).

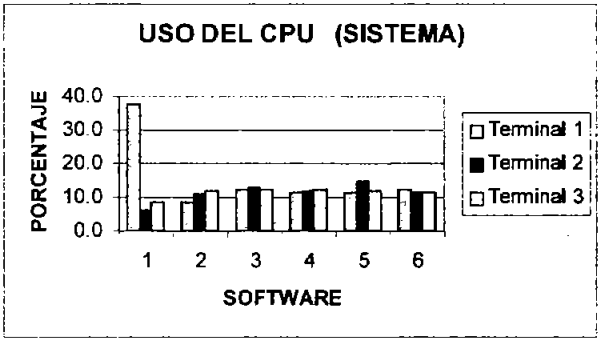
La prueba consistió en ejecutar una aplicación en la primer terminal, en este caso el navegador, luego se anotaron los valores de los recursos actuales (procesador, memoria y swap); posteriormente se hizo lo mismo con la segunda y tercer terminal. Después que se hicieron las mediciones correspondientes al primer programa, se prosiguió a ejecutar el procesador de texto, continuando con la misma mecánica se ejecutaron las aplicaciones restantes; hasta que al final se recopiló la información de cuanto fue el consumo de recursos por las tres terminales y a su vez para las 6 aplicaciones.

La siguiente tabla muestra los datos arrojados en la prueba antes descrita:

SOFTWARE	CPU (%)						MEM (Kb)			SWAP (Kb)		
	Term 1		Term 2		Term 3		Term 1	Term 2	Term 3	Term 1	Term 2	Term 3
	SIST	USR	SIST	USR	SIST	USR						
Navegador	37.6	20.5	6.0	15.9	8.7	19.7	371040	371120	365912	0	0	0
Proc. de texto	8.7	19.7	11.0	12.2	12.1	13.6	372036	371980	372216	0	0	0
Hoja de cálc.	12.2	15.2	12.9	11.6	12.5	14.7	372200	371180	371464	0	0	1132
Xcircuit	11.5	15.0	12.1	13.6	12.5	19.1	372152	372092	370964	1132	1132	1104
Tkgate	11.4	14.5	14.6	14.8	11.7	19.6	371568	371520	371508	1104	1104	1104
Bluehdl	12.4	14.8	11.4	16.8	11.2	16.1	371412	371792	371712	1104	1104	1104

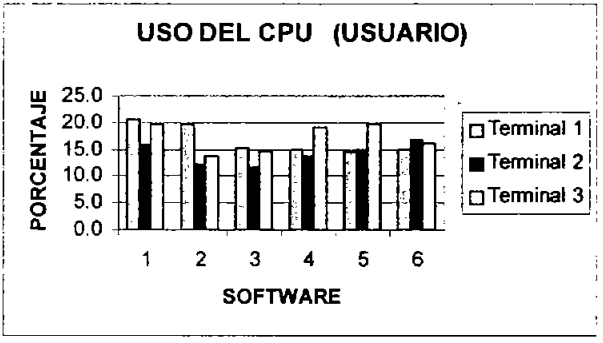
Tabla 5-2 Resultados obtenidos de las pruebas realizadas

Como puede observarse, al ejecutar el navegador en la primer terminal, se obtuvo un alto porcentaje en el uso del procesador; esto se debió a que el sistema le dio prioridad máxima a esta aplicación e incremento la actividad de procesamiento, pero después de cierto tiempo el sistema se volvió a estabilizar gracias a las características del planificador de procesos antes mencionadas.



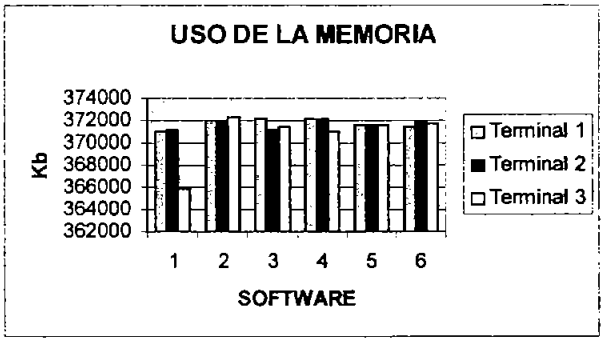
Gráfica 5-3 Uso del CPU (sistema) con las tres terminales

En el caso del uso del procesador por parte de las aplicaciones de usuario, se observa que no hubo mucha variación en el porcentaje; ya que se tiene la ventaja de que se ejecutaron las mismas aplicaciones para las tres terminales, optimizando en gran medida el uso de este dispositivo. Además con estas pruebas también fue posible comprobar la eficiencia de Linux para manejar la concurrencia en las aplicaciones.



Gráfica 5-4 Uso del CPU (usuario) con las tres terminales

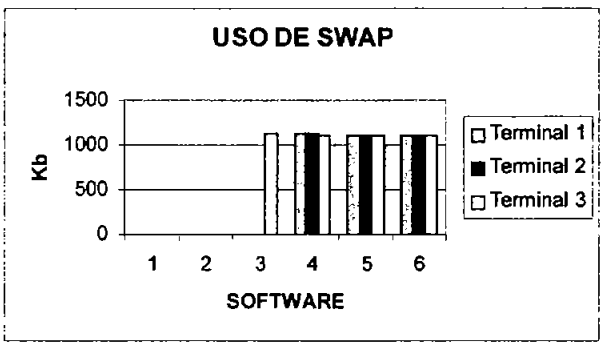
Con respecto a la memoria, se puede notar que al ejecutar cada uno de los programas ésta prácticamente se ha agotado, pero el sistema aún tiene un buen funcionamiento ya que de cierta forma llega a un punto de estabilización que impide que el sistema se colapse.



Gráfica 5-5 Uso de la memoria con las tres terminales

Sin embargo cuando la memoria física se ha terminado, Linux comienza a hacer uso de la memoria swap, sin ella las terminales se hubieran colapsado cuando la tercer terminal hiciera la petición para ejecutar el tercer programa; con esta memoria auxiliar es posible seguir trabajando y tener soporte para ejecutar tres programas más con las tres terminales funcionando adecuadamente.

La memoria swap es importante pero no puede reemplazar a una memoria física, pero puede permitir al sistema ejecutar aplicaciones más grandes guardando en disco duro aquellas partes de código que no se utilizan.



Gráfica 5-6 Uso de swap con las tres terminales

Considerando que se empezó a utilizar la memoria swap a partir del tercer programa y se cuenta con 384 MB de RAM en el servidor, se puede concluir que con un promedio de 3 aplicaciones corriendo, se necesita aproximadamente 100 MB de memoria por cada terminal; de esta manera si en un futuro se requiere expandir la red, se recomienda que por cada terminal que se incorpore, se incremente proporcionalmente el tamaño de la memoria, por ejemplo para 10 terminales se deberá tener aproximadamente 1 GB de RAM.

En cuanto al procesador, se recomienda que para más de 5 terminales, su velocidad sea mayor o igual a 1 GHz, con esto se garantiza un buen desempeño y confiabilidad de la red, solo si se desea utilizar una computadora personal como servidor.

Pero si se desea implementar este proyecto de una manera más formal, es decir, más de 15 terminales, lo más adecuado es invertir en un servidor, ya que es un equipo diseñado para estar conectado las 24 horas del día durante todo el año, el cual tiene que soportar fuertes picos de actividad, y debe tener un alto rendimiento en ejecución de aplicaciones para que los alumnos no vean afectada la velocidad y que cuente básicamente con las siguientes características, tomando como referencia las pruebas que se realizaron.

Procesador: Intel Pentium IV
Frecuencia: 3 Ghz
Memoria: 4GB
Disco Duro: 160 GB

La empresa Hewlett Packard ofrece un servidor económico de $17 000, con estas características realmente no es mucha la inversión tomando en cuenta que de esta forma se podrá tener varias 486 funcionando como estaciones de trabajo, si se invirtiera esta cantidad en computadoras medianamente potentes solo se lograría adquirir aproximadamente 4, sin tomar en cuenta la cantidad faltante por costos de licencias. Qué mejor inversión que la de adquirir este servidor y poner a funcionar esas viejas 486 brindando una potencia aceptable y sin gastos por licencias de software.

5.2 Seguridad

Con el fin de garantizar que la red de terminales gráficas funcione adecuadamente y que los recursos del servidor estén disponibles en cualquier momento, se pretende dar algunas ideas generales de los riesgos existentes y las medidas de seguridad para prevenir los posibles ataques al sistema.

La meta de la mayoría de los ataques a un sistema operativo es conseguir un acceso con privilegios de root, lo que garantiza un control total del sistema; primero se

intentará conseguir acceso como usuario normal e ir incrementando los niveles de privilegio utilizando las posibles vulnerabilidades del sistema.

Se puede decir que con el simple hecho de intentar violar la privacidad de la información, obteniendo privilegios que no le corresponden a un determinado usuario, haciendo un uso desmedido de los recursos o modificando información legítima contenida en una máquina, como puede ser el contenido de una página web o una base de datos; éstas son algunas formas de poner en riesgo el correcto funcionamiento del sistema.

En todo sistema la seguridad simplemente es fundamental, se debe tomar en cuenta que en un ambiente LTSP pueden haber muchos usuarios conectados al servidor al mismo tiempo; por lo que el sistema operativo del servidor tendrá la obligación de proteger a unos usuarios frente a otros y protegerse así mismo.

En primer lugar se debe tener en cuenta que ningún sistema es completamente seguro, solo lo será cuando no este conectado a la red o simplemente este apagado. Desde esta perspectiva se puede partir de que lo único que se puede hacer es aumentar la dificultad para que alguien pueda comprometer la seguridad del sistema.

Por tal motivo, es conveniente establecer algunas políticas que fijen el nivel de seguridad que se requiere, partiendo desde las políticas de administración del servidor, así como algunas políticas de uso para los alumnos, los cuales serán los que potencialmente podrán comprometer la seguridad de la red del laboratorio.

Lo que se desea es garantizar que el sistema permanezca funcionando de forma adecuada con comunicaciones seguras, donde nadie pueda obtener o modificar información a la que no tiene derecho. Una buena planeación ayuda bastante y antes de asegurar el sistema se debe determinar contra qué quiere protegerse y qué riesgos acepta.

Se pueden mencionar algunos requisitos de seguridad:

- Disponibilidad: Mantener la información y los recursos garantizando que no se limite el acceso autorizado a la información y el correcto funcionamiento de los recursos.
- Integridad: Requiere que la información sólo pueda ser modificada por las entidades autorizadas.
- Autenticidad: La información debe permanecer protegida ante falsificaciones, requiere de mecanismos de identificación asegurando que las comunicaciones se realizan entre entidades legítimas.
- Confidencialidad: Pretende evitar la difusión no autorizada de la información, requiere que ésta sea accesible únicamente por las entidades autorizadas.

Además en Linux se deben proteger ciertos archivos que contienen información de los usuarios (/etc/passwd, /etc/shadow), los archivos de configuración del sistema (los contenidos en etc.), el acceso al sistema y la correcta utilización de los recursos.

En muchas ocasiones los peores ataques al sistema los realiza el administrador, cuando por descuido, ignorancia o exceso de confianza altera el sistema de forma inapropiada.

Para evitar estos problemas solo hay que seguir ciertas normas de prevención como:

- No usar la cuenta de root como predeterminada, intentar primero cualquier acción como un usuario normal, si no tiene permiso puede hacer uso del comando "su" si es necesario.
- Ejecutar los comandos de forma segura verificando previamente la acción que se va a realizar.
- Activar la opción (-i) para actuar de forma interactiva con ciertos comandos que la admiten.

Al administrar un sistema Linux se deben tomar ciertas medidas que garanticen un mínimo de seguridad y funcionalidad, en primer lugar se puede limitar el espacio asignado a los usuarios ya que un posible ataque a cualquier sistema es intentar consumir todo el espacio del disco duro, una medida de seguridad es separar el árbol de directorios en diversas particiones. Pero esto puede no ser suficiente y por eso el núcleo del sistema proporciona la posibilidad de controlar el espacio de almacenamiento por grupo o usuario.

Es muy probable que muchas de las intrusiones puedan ser realizadas por los alumnos, por lo que se debe proteger al sistema de operaciones accidentales debidas a descuidos o ignorancia de parte ellos.

Una norma básica de seguridad radica en la asignación a cada usuario sólo de los permisos necesarios para poder cubrir las necesidades de su trabajo sin poner en riesgo el trabajo de los demás.

Para esto Linux dispone de todas las características de los sistemas Unix, ya que se mantiene un control de acceso a los usuarios donde se verifica login y password; por lo que se debe asegurar que los usuarios utilicen claves sólidas que cumplan con las siguientes características:

- No debe ser una palabra conocida
- Debe de contener letras, números y caracteres especiales
- Deben ser fáciles de recordar y difíciles de adivinar.

Para mantener un sistema seguro pero funcional, se tienen que otorgar los permisos adecuados para que los usuarios puedan tener acceso solo a los recursos autorizados, considerando los permisos de propietario, grupo y otros. Afortunadamente Fedora Core tiene por defecto los valores que son suficientes para mantener el sistema seguro.

Una vez que se han visto las características generales de seguridad, lo que queda es aplicar el sentido común, se tiene que ver la situación actual y responder a una serie de preguntas:

- ¿Qué se quiere proteger?
- ¿Qué valor tiene lo que se quiere proteger?
- ¿Qué costo tiene la seguridad?
- ¿De quién se quiere proteger?
- ¿Cuáles son los puntos débiles del sistema?

Después de responder estas preguntas, el administrador del sistema obtendrá un esquema elemental de la situación actual y de las medidas de seguridad básicas que debe tomar.

5.3 Trabajo futuro

Se considera que lo expuesto en esta tesis da la pauta para continuar la investigación en el área del software libre, en especial de las redes de terminales gráficas; a continuación se darán algunas ideas que pueden tomarse como base para trabajos futuros y para mejorar los resultados obtenidos:

- Se propone en primer lugar la mejora de las características del servidor para que éste pueda dar soporte a un mayor número de terminales. De acuerdo a las pruebas que se efectuaron, se pudo observar que la memoria es determinante en el rendimiento de la red, por lo que se recomienda tener 1 GB si se requiere expandir la red de 3 a 10 terminales. También es recomendable utilizar como medio de almacenamiento un disco duro SCSI en lugar de un disco IDE, ya que permite transferencias de datos con una mayor velocidad y de manera más confiable; éste aspecto es importante debido a que en un momento determinado todas las terminales pueden estar utilizando este recurso, por lo que el rendimiento de la red puede disminuir.

- Continuando con la parte de optimización de la red, se propone la implementación del proyecto PXES, que es una alternativa al LTSP, pero a diferencia de éste último permite aprovechar los recursos de las terminales con mejores capacidades, lo que

implica que se puedan correr aplicaciones localmente utilizando su memoria y disco duro, disminuyendo considerablemente la carga al servidor. Cabe mencionar que las terminales con menores capacidades como las 486, pueden seguir con el esquema del LTSP, ya que como se pudo observar en este trabajo, es el óptimo para este tipo de computadoras.

- Una posible aplicación de la red implementada en el laboratorio, es la creación de un cluster; esta arquitectura permite utilizar los recursos de las terminales para ejecutar operaciones de cálculo complejas que requieren mayor tiempo de procesamiento. El software que se propone utilizar es Open Mosix, el cual incorpora al kernel de Linux un potente algoritmo de balanceo de cargas para dividir el trabajo de procesamiento entre todos los equipos. Se debe considerar que las terminales deben poseer características de procesamiento y memoria aceptables para que el tiempo en resolver un problema específico sea mínimo.

- Otro proyecto interesante que se propone es el de la realización de una página Web que permita controlar remotamente dispositivos electrónicos conectados a las terminales, por ejemplo circuitos basados en microcontroladores. Esto permitiría entre otras cosas, monitorear el comportamiento de algunos procesos ligados a estos circuitos los cuales tienen potencialmente muchas aplicaciones. Con esto se pretende ofrecer una infraestructura que permita impulsar a los estudiantes a que desarrollen proyectos de electrónica cada vez más novedosos y funcionales.

- Finalmente, aunque es un proyecto independiente a la institución; con la experiencia adquirida en el desarrollo de esta tesis, se pretende implementar el LTSP en un cybercafe, se propone invertir en un servidor que pueda dar soporte a más de 15 computadoras 486 equipándolas con software de todo tipo, así como de Internet. La renta de los equipos de computo sería realmente mínima tomando en cuenta que pueden proporcionar los mismos resultados que si se utilizaran computadoras Pentium III, con la diferencia de brindar el mismo servicio pero a un menor precio, esto debido al ahorro de licencias, mantenimiento de los equipos, etc., y a su vez las personas se familiarizan con el uso de Linux. Posteriormente se pretende crear un servidor de música para dichas terminales, así como un servidor Samba que permita compartir recursos y archivos con diferentes plataformas de Windows (95, 98, NT, 2000, Me) y de esta manera dichas plataformas puedan acceder a los archivos Linux. El reactivar estas computadoras 486 para darle nuevas aplicaciones, y alargar su ciclo de vida; permitirá tener un negocio rentable con inversiones mínimas.

CONCLUSIONES

La idea básica del proyecto fue rescatar computadoras relativamente obsoletas como las 486 y pentium brindándoles una nueva apariencia de procesamiento gracias a la dependencia hacia un servidor, el cual les brinda sus recursos para lograr obtener unas terminales totalmente funcionales que cuentan con una buena velocidad de procesamiento, acceso a internet, procesadores de texto, software de electrónica, hasta diseños en 3D, etc., y todo bajo un ambiente gráfico totalmente amigable.

Se pudo corroborar todo esto ya que las terminales fueron probadas con éxito en el laboratorio, al principio se experimentó la facilidad de instalación del sistema operativo y una vez que se implementó el LTSP, el mantenimiento así como la actualización del software se efectuaron de una forma sencilla. El sistema realmente demostró su estabilidad y confiabilidad, nos ofrece la libertad de poder ajustarlo a nuestras necesidades, aunado a muchas otras ventajas y beneficios que se encontraron al implementar este proyecto.

También se logró recopilar una gran variedad de software de electrónica totalmente gratuito, el cual ofrece nuevas posibilidades de desarrollo para los alumnos. Además éste tipo de software permite un gran ahorro en licencias, así como de hardware adicional debido a que estas aplicaciones no necesitan de mayores recursos para funcionar adecuadamente.

Además este tipo de terminales bajo Linux no recibe tantos ataques de virus como en Windows, podemos decir que este tipo de sistema es realmente recomendable no sólo para instituciones públicas o pequeñas empresas, sino también para grandes empresas que deseen adquirir un sistema potente, seguro, y muy económico. No requiere de un gasto adicional para capacitación ya que Linux es muy intuitivo por su ambiente gráfico.

Es verdad que en Linux existen ciertos problemas con algunos modems, tarjetas de video e impresoras; donde para obtener una buena configuración se requerirá de varias horas, pero también tenemos la opción de que Linux permite ejecutar muchos programas de Windows mediante emuladores.

Con respecto al tiempo de implementación no hay mayor problema, ya que al hallar la configuración adecuada para una terminal las otras configuraciones son muy similares, esto reduce en gran medida el tiempo de implementación ya que solo es cuestión de configurar ciertas partes.

Al hacer uso de la protección de la memoria entre procesos, de manera que ninguno de ellos detenga el sistema como suele suceder usualmente en Windows, se obtiene una estabilidad determinante para el sistema así mismo cuenta con una mejor

optimización de los recursos de la computadora brindándonos la confiabilidad de realizar varios procesos sin problemas; es un sistema mucho más robusto sobre todo para aplicaciones de alta demanda.

Se optó por utilizar el proyecto LTSP debido a que nos permite aprovechar los recursos de computadoras de baja capacidad, si se contará con computadoras medianamente más potentes se pudó haber utilizado por ejemplo el proyecto de PXES ya que permite aprovechar los recursos de las terminales con mejores capacidades, lo que implica que se puedan correr aplicaciones localmente utilizando su memoria y disco duro, disminuyendo considerablemente la carga al servidor. Es por eso que promovemos el uso de un servidor más potente para lograr atender las peticiones de las terminales ya que éstas cuentan con pocos recursos, y dependen de las características del servidor.

Comprobamos que se eligió la mejor opción para reutilizar a las 486 disponibles, logrando darles nuevamente la funcionalidad que se buscaba así como nuevos campos en los que pueden ser utilizadas estas terminales bajo el concepto del LTSP.

REFERENCIAS

RAYA José Luis, RAYA Cristina, Redes Locales, Ed. Alfaomega Ra-Ma, 2002

FLORIANO Blanco Lino, GALAN Galán Susana, Al día en una hora Linux, Ed. Anaya Multimedia, 1997

CARLING M. DEGLER Stephen DENNIS James, Administración de sistemas Linux, Ed. Prentice Hall

PERRY Hekman Jessica, Linux in a nutshell. A desktop quick reference, Ed. Perry O'reilly, 1997

HUSAIN Kamran, PARKER Tim, Linux unleashed, Ed. Sams published, 1995

CARD Remy, DUMAS Erick, MEVEL Frank, The Linux kernel book, Ed. John Wiler & Suns, 1998

BLANCO J. Vicente, Linux Instalación, administración y uso del sistema, Ed. Ra-ma, 1996

NEGUS Christopher, La biblia de Red Hat Linux 7, Ed. Anaya multimedia, 2001

ESPINOSA H. Juan Carlos, Red Hat Linux 7.0 Instalación y configuración básica, Ed. Alfaomega, 2001

BENTSON Randolph, Inside Linux. A look at operating system development, Ed. Specialized System Consultants, Inc. 1996

LES Fred, DERFLER J. Jr. Frank, Guía de Novel Netware, Ed. Anaya multimedia, 1993

STAN Kelly – Bootle, Como usar Unix sistema V, versión 4.0, Ed. Noriega editores, 1993

MEDIAVILLA Manuel, Seguridad en Unix, Ed. Ra-ma, 1998

WELSH Matt, KAUFMAN Lar, Running Linux, Ed. O'reilly & associates, INC, 1996

STEFAN Strobel, VOLKER Elling, Linux unleashing the workstation in your PC, Ed. Springer, 1997

Página web de la FSF, Sección de Filosofía http://www.fsf.org/philosophy/free-sw.es.html

Página web del Proyecto GNU, Sección de Historia
http://www.gnu.org/gnu/thegnuproject.es.html

Página web del Proyecto GNU, Sección de Licencias
http://www.gnu.org/copyleft/gpl.html

Página web del Proyecto LTSP, Sección de Documentación
http://www.ltsp.org/documentation/ltsp-3.0-4-es.html

Página web de creación de imágenes de arranque http://www.rom-o-matic.net

Página web de Fedora Core http://fedora.redhat.com/download/

Página web del proyecto K12LTSP, Sección de Instalación
http://k12ltsp.org/install.html

Página web acerca de estaciones gráficas diskless
http://www.logiclinux.com/soluciones/diskless/

Página web acerca de experiencias con el LTSP http://www.cignux.org.ar/otkraus.htm

Página web de referencia para implementar un cybercafe
http://webmaster.bankhacker.com/ayuda-linux/consulta/Instalar+Cybercafe/

GLOSARIO

ASCII: (American Standard Code for Information Interchange). Es el código estándar de conjunto de caracteres que cualquier computadora puede entender, usado para representar las letras latinas, en mayúsculas, minúsculas, números, puntuación, etc. Hay 128 códigos estándar ASCII, cada uno de los cuales puede representarse por un número binario de 7 dígitos. Sin embargo, otros conjuntos de caracteres como Latin-1 están comenzando a usarse.

Broadcast: Es un tipo especial de dirección IP usada para enviar un mensaje a todos los nodos de una red.

Cliente: Programa que se usa para contactar y obtener datos de un programa de servidor localizado en otra computadora, generalmente a gran distancia. Cada programa cliente está diseñado para trabajar con uno o más tipos de programas servidores específicos, y cada servidor requiere un tipo especial de cliente.

Demonio: Es un proceso independiente encargado de llevar a cabo una tarea sin la intervención del usuario.

Dirección IP: Dirección que identifica a cada computadora en Internet utilizando una fila de cuatro conjuntos de números separados por puntos.

Diskless: PC o estación de trabajo conectado a una LAN sin disco duro ni disquetes y que depende del almacenamiento en disco de un servidor de archivos conectado a la misma LAN.

DNS: (Domain Name System). Es un sistema que traduce un nombre de dominio a un número IP, y se emplea para facilitar el manejo de direcciones ya que es más fácil recordar un nombre que un grupo de números.

EMACS: (Editing MACroS o Extensible MACro System). Es un editor muy popular para Unix y otros sistemas operativos.

Ethernet: Es la tecnología de red de área local más ampliamente utilizada. Es un estándar de la industria muy extendido, desarrollado originalmente por Xerox y formalizado en 1980 por DEC, Intel y Xerox. Las redes Ethernet transmiten datos a 10/100/1000/ Mbps utilizando un protocolo especificado.

FTP: (File Transfer Protocol). Permite transmitir archivos sobre Internet entre una máquina local y otra remota.

Host: Es una computadora anfitriona que permite a varios usuarios correr aplicaciones o programas diferentes.

Runlevel: Es el estado general de un sistema operativo tipo Unix, el cual es definido por el proceso init.

Script: Es un programa o secuencia de instrucciones que son interpretadas o ejecutadas por otro programa.

Sistema de archivos: Un sistema de archivos es el método con el que la información es introducida en un disco duro. Sistemas operativos distintos generalmente usan sistemas de archivos diferentes.

TCP/IP: (Transmisión Control Protocol/Internet Protocol) familia de protocolos que hacen posible la interconexión y trafico de red en internet.

Telnet: Remote login es un protocolo que forma parte de la suit de protocolos TCP/IP y que permite conectar una terminal remota a su computadora host.

VESA: Es un estándar de modos de video para tarjetas VGA y superiores, que permite programar drivers compatibles con todas las tarjetas graficas que cumplan estas normas, independientemente del chip que incorporen.

CONCEPTOS GENERALES

OBJETIVO ESPECÍFICO

Proporcionar los conocimientos básicos mediante conceptos, definiciones y aspectos técnicos, que son requeridos previamente y durante el desarrollo e implementación del LTSP.

2

INTRODUCCIÓN

Conforme las computadoras se van acoplando a la vida diaria, cada vez se usan más para resolver los problemas, y si una sola computadora resulta muy valiosa por su capacidad para procesar información sin necesidad de influencia externa, cómo sería si dicha computadora logrará comunicarse con otras, imagine las capacidades adicionales que estarían disponibles si al conectar las computadoras lograrán desplegar la información contenida en otras máquinas. Considere los beneficios que se lograrían si un grupo de trabajo pudiera acceder al mismo tiempo a un archivo.

Una red hace posible esto y mucho más, es por eso que es importante conocer los diferentes componentes de una red, y que mejor si se pueden encontrar opciones que optimicen el rendimiento de las máquinas tomando en cuenta que no se tenga que invertir mucho.

La instalación de una red no tiene por qué suponer un gran desembolso de dinero para adquirir las computadoras cliente, podemos aprovechar los viejos i486/66 como terminales gráficos. Sólo será necesario que dispongan de tarjeta de red, pudiendo prescindir de unidades de disco duro, disqueteras o cd-rom. Computadoras mejor equipadas no tienen por qué mostrar necesariamente un mayor rendimiento, teniendo en cuenta que las aplicaciones son ejecutadas en el servidor. Los requerimientos del servidor dependerán del número de terminales que atenderá, siendo importante no sólo el procesador sino también la memoria RAM instalada en el servidor.

Por otro lado, el pronosticar a cuántas terminales puede crecer una red es una tarea necesaria para determinar si la alternativa propuesta puede ser viable, o a corto plazo será inoperable. Inicialmente puede escogerse una red de alto rendimiento, pero esto implica un costo más alto, tanto en la instalación como en su mantenimiento.

2.1 Redes

Las redes locales son una alternativa cada vez más viable para medianas y grandes organizaciones es por eso que se necesita dar un panorama para entender como funcionan y así elegir la red que éste de acuerdo a las necesidades y recursos de cada organización.

2.1.1 La red de área local

La red de área local nació con la necesidad de conectar las PC's a fin de compartir información, éstas se pueden clasificar de acuerdo a su distribución geográfica o por el servicio que ofrezcan. Cuando los dispositivos de una red se encuentran en un área mínima como un edificio o una escuela se está hablando de una red de área local, éste sistema de comunicación de alta velocidad permite que las PC's que se encuentran cerca puedan comunicarse entre sí. A su vez la red consta de hardware y software de red y sirve para interconectar las PC's que están aisladas. Esta conexión puede ser directa, por cable coaxial, indirecta o por módem, entre otros. Los diferentes dispositivos que la componen se comunican por una serie de reglas que se conocen como protocolo.

Aunque apareció desde 1983, la LAN ha continuado evolucionando hasta llegar a ser una parte integral de la conectividad de las PC's, además de ser confiable.

El proceso de incorporar una PC a una LAN consiste en la instalación de una tarjeta de interfaz de red (NIC) en cada computadora y esta se conecta con un cable especial para red. El último paso para implantar una LAN es instalar en cada PC un software conocido como sistema operativo de red (NOS).

Si se dividiera una red en sus componentes más simples, se tendrían dos partes, la física: el cableado, las tarjetas de red, las computadoras y demás equipo que utiliza la red para transmitir los datos. La otra parte es la disposición lógica de esos componentes físicos: las reglas que permiten a los componentes físicos trabajar en conjunto. En general a estas dos partes se les conoce como hardware y software de red.

2.2 Hardware de red

Las redes están conformadas por muchos elementos de hardware diferentes que deben trabajar juntos para crear una red funcional, estos dispositivos los fabrican por lo general varias compañías, por lo tanto es necesario que haya comunicación entre los fabricantes en relación con la manera en que los componentes de la red trabajan e interactúan entre sí, esto con el fin de lograr una mayor compatibilidad entre ellos. La conectividad del hardware de red depende de las características con las que fue diseñado y también de la manera en que se implementa; ya que existen múltiples variantes en cuanto a la instalación y configuración de los diferentes elementos de la red.

2.2.1 Topologías

Los nodos de red necesitan estar conectados para comunicarse y a la forma en que están conectados se le llama topología, que consta de una topología física y una lógica. La física se refiere a la manera en que los nodos están conectados unos con otros y la lógica es el método que se usa para comunicarse con los demás nodos, es decir; la ruta que toman los datos dentro de la red para llegar a los diferentes nodos. Las topologías física y lógica pueden ser iguales o diferentes.

Cada red local usa un cable para llevar la información, este cable debe controlar el flujo de la información en la red de tal forma que los mensajes puedan ser transferidos de una manera confiable. En la mayoría de las topologías las señales son enviadas en todas direcciones desde el equipo emisor, cada dispositivo tiene su propia dirección asignada, y el software se utiliza para programar un dispositivo que acepte mensajes con dirección única, e ignora a todos los demás.

Las 3 topologías de red estándar son de bus, estrella y anillo; también hay combinaciones de más de una topología.

2.2.1.1 Topología de bus

En una topología de bus, cada computadora está conectada a un segmento común de cable de red. El segmento de red se coloca como un bus lineal, es decir, un cable largo que va de un extremo a otro de la red, y al cual se conecta a cada nodo. El cable puede ir por el piso, por las paredes, por el techo, o puede ser una combinación de éstos, siempre y cuando el cable sea un segmento continuo.

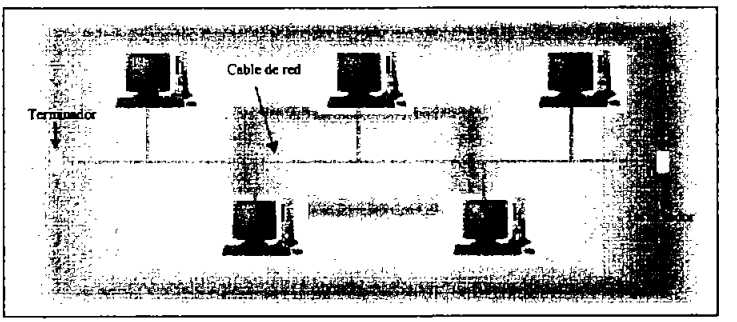


Figura 2-1 Topología de bus

2.2.1.2 Topología de estrella

En una topología de estrella, cada computadora está conectada a un concentrador (o hub) ubicado centralmente. El concentrador es un dispositivo de hardware con varios puertos, al cual se puede conectar un cable de red en cada uno de ellos, donde cada máquina tiene un enlace exclusivo con el hub.

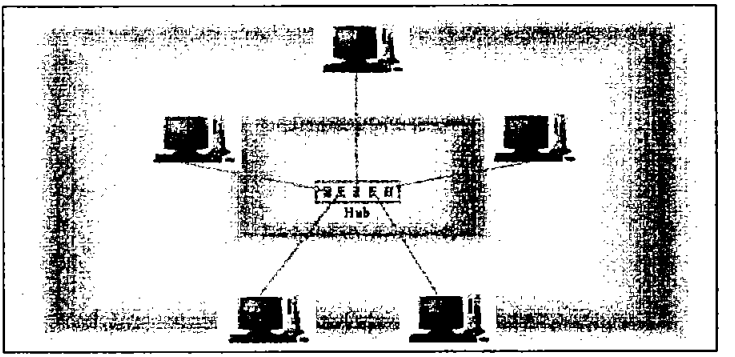


Figura 2-2 Topología de estrella

2.2.1.3 Topología de anillo

En una topología de anillo, cada computadora se conecta en forma circular a la red. Las topologías de anillo casi siempre son lógicas con topología física de estrella. La topología física muestra que cada computadora se conecta a un dispositivo central y parece una estrella. La ruta seguida por los datos de una computadora a otra muestra que la topología lógica es de anillo.

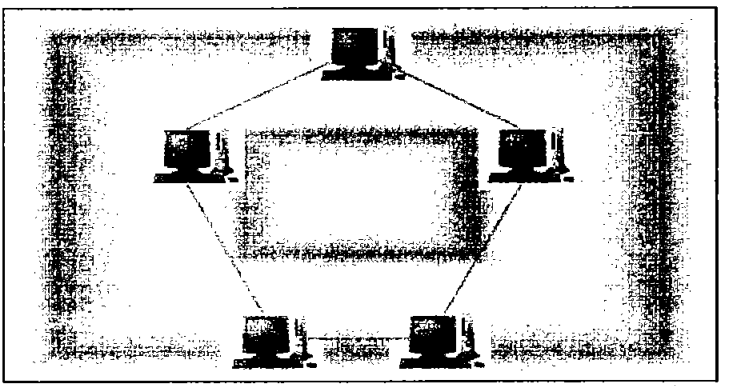


Figura 2-3 Topología de anillo

2.2.1.4 Topología de árbol

Una topología de árbol es la combinación de las topologías de bus y de estrella. Muchos concentradores de las redes Ethernet con topología física de estrella también tiene un conector en la parte trasera que enlaza al concentrador a una red de topología física de bus.

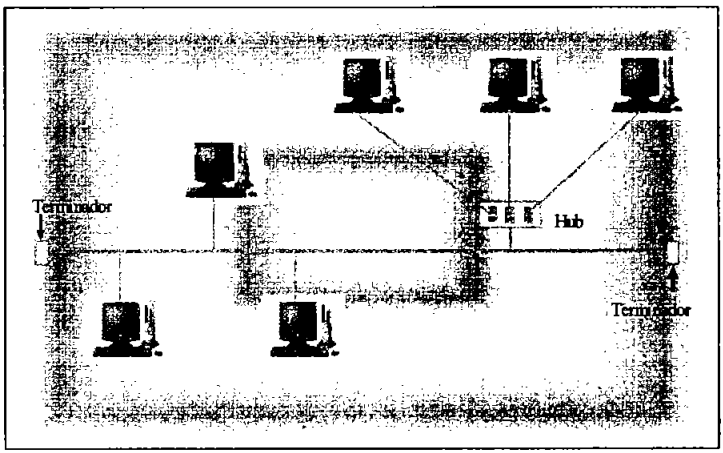


Figura 2-4 Topología de árbol

2.2.2 Estándares de redes

Por suerte se han creado estándares que definen la forma de conectar componentes de hardware en las redes y el protocolo (o reglas) de uso cuando se establecen comunicaciones por red. Donde el protocolo define la forma en que deben efectuarse las comunicaciones de las redes.

Un estándar es la especificación de red adoptada, incluye reglas que se refieren al tipo de componentes que deben usarse, a la manera de conectar los componentes, así como a los protocolos de comunicación que hay que emplear.

Los 3 estándares más populares que se utilizan son:

- Ethernet
- ARCnet
- Token Ring.

2.2.2.1 Ethernet

Ethernet también se le conoce como IEEE 802.3, es el estándar más popular para las LAN que se usa actualmente y transmite datos a través de la red a una velocidad de 10 Mbps (megabits por segundo). Este estándar usa un método de acceso al medio conocido como Acceso Múltiple con Detección de Portadora y Detección de Colisiones (CSMA / CD). Antes de que un nodo envié algún dato a través de la red, primero escucha y detecta si algún otro está transfiriendo información.

De no ser así, el nodo transfiere la información a través de la red. Los otros nodos escucharán y el nodo seleccionado recibirá la información. En caso de que dos nodos traten de enviar datos por la red al mismo tiempo, cada nodo se dará cuenta de la colisión y esperará una cantidad de tiempo aleatoria antes de volver a hacer el envío.

Aunque CSMA / CD es una forma rápida y eficiente para transmitir datos, una red muy cargada podría llegar al punto de saturación. Conforme más nodos tratan de transmitir información por la red, más aumenta la posibilidad de colisiones y se reduce de modo importante la eficiencia de la red. Sin embargo, con una red diseñada adecuadamente, la saturación rara vez es preocupante.

Existen 3 estándares de Ethernet, 10BASE5, 10BASE2 y 10BASE-T, que definen el tipo de cable de red, las especificaciones de longitud y la topología física que debe utilizarse para conectar nodos en la red.

2.2.2.1.1 10BASE5

También llamado Ethernet estándar, Thick Ethernet o Thicnet, tiene un estándar de topología física de bus que consiste en un segmento de cable de red con terminadores en los extremos. Los terminadores incluyen una resistencia que disipa la señal de la red y no permite que se refleje de regreso al cable de red. Su NIC está conectada a un transmisor-receptor (transceptor) externo por medio de un cable de suspensión. El transceptor está conectado al segmento de cable Thicnet y actúa para transmitir y recibir datos de la red entre la computadora y la red.

Es relativamente difícil trabajar con Thicnet, en comparación con las otras dos disposiciones, 10BASE2 y 10BASE-T. Sin embargo debido a que fue la única disponible durante un tiempo, Thicnet se encuentra todavía en varias instalaciones.

2.2.2.1.2 10BASE2

También llamado Thinnet, Thin coax o Thin Ethernet, éste se instala por medio de una topología física de bus, que consiste en segmentos de cable con terminaciones en

cada extremo. La NIC de cada computadora está conectada directamente al segmento de cable Thinnet; lo que descarta la necesidad de un transceptor externo.

Thinnet fue muy popular en negocios e instalaciones pequeños, debido a que es el método menos caro para poner en servicio una red Ethernet ya que se emplea una pequeña cantidad de nodos. Además es menos susceptible a la interferencia eléctrica que el par trenzado. Una desventaja, es que si hay una ruptura en cualquier parte del cable, dejará de funcionar toda la red. Así mismo puede ser ardua la búsqueda de fallas causadas por un problema de cable.

2.2.2.1.3 El 10BASE-T

Éste a diferencia del Thick o Thin Ethernet, se instala por medio de una topología física de estrella donde cada nodo se conecta a un hub y la NIC de cada computadora se conecta al concentrador por medio de un segmento de cable de red.

Con frecuencia se prefiere el 10BASE-T sobre el 10BASE2, por su flexible topología de estrella; ya que la ruptura en el cable de una red con 10BASE-T solo desactivará a la computadora que esté al extremo de la línea rota, en vez de toda la red como sucede con el otro estándar. El 10BASE-T es más barato para redes pequeñas aunque requiere un concentrador adicional; sin embargo el cable de par trenzado que se emplea en 10BASE-T es menos caro, por lo que entre más nodos se añadan, el gasto adicional de un concentrador será menor en comparación con el costo de utilizar el cable Thinnet.

2.2.2.2 Token Ring

Llamado IEEE 802.5, este estándar opera a una velocidad de 4 Mbps o 16 Mbps, emplea topología lógica de anillo y una topología física de estrella. La NIC de cada computadora se conecta a un cable y éste a su vez a un hub llamado unidad de acceso a multiestaciones (MAU). Token Ring se basa en un esquema de paso de señales (token passing) es decir, se transmite una señal a todas las computadoras de la red donde la computadora que esté en posesión del token tiene autorización para transmitir su información a otro nodo.

Cuando termina, el token pasa a la siguiente computadora, si ésta tiene que enviar información acepta el token y procede a enviarla; en caso contrario, el token pasa a la siguiente computadora del anillo y el proceso continúa.

La MAU se salta automáticamente un nodo de red que no esté encendido. Sin embargo, dado que cada nodo de una red Token Ring examina y luego retransmite cada token, un nodo con mal funcionamiento puede hacer que deje de trabajar toda la red.

2.2.2.3 ARCnet

ARCnet es un estándar similar al IEEE 802.4 y en 1992 ANSI lo reconoció como estándar formal. Soporta una velocidad de transferencia de datos de 2.5 Mbps, usa una topología lógica de bus y una ligera variación de la topología física de estrella.

Cada nodo está conectado a un concentrador pasivo o a uno activo. Un concentrador pasivo no recibe potencia eléctrica y sirve para distribuir la señal de la red a cortas distancias, en cambio un concentrador activo sí recibe potencia eléctrica y también amplifica la señal de la red para permitir que la red cubra distancias más largas.

La NIC en cada computadora está conectada a un cable que a su vez esta conectado a un concentrador activo o pasivo.

2.2.3 Adaptadores de red

Para comunicarse con el resto de la red, cada computadora debe tener instalada una tarjeta de interfaz de red (NIC), la cual se conecta a otros nodos por medio del cable de red empleando y los protocolos adecuados.

Aunque la mayor parte de los adaptadores de red se instalan en el interior de la computadora, algunos son de instalación externa. El tipo de adaptador de red que se compre determinará la velocidad de transmisión de datos, el tamaño de las unidades de mensaje (paquetes), así como la topología de la red, por lo que hay que estar seguro de obtener el adaptador de red adecuado para la topología que se quiera usar.

Para que un adaptador funcione como interfaz entre la computadora y la red, debe satisfacer 2 criterios:

1. Debe ser del tipo adecuado para la red con la que se va a conectar, utilizando los protocolos correspondientes para comunicarse con el resto de la red y tener el conector apropiado para el cable.

2. Debe tener el conector apropiado para la ranura de expansión de la computadora ya que en ocasiones los equipos de diferentes fabricantes tienen distintos tipos de ranuras de expansión.

La mayor parte de las computadoras tienen ranuras de expansión ISA, las cuales son de 8 o 16 bits. Actualmente las computadoras cuentan con ranuras de expansión, como el bus local VESA o el bus local PCI que permiten que pasen los datos en la ranura de bus local a mayor velocidad que las ranuras ISA estándar.

2.2.3.1 Adaptadores Ethernet

En la actualidad, casi todas las redes se basan en el estándar Ethernet y por lo tanto lo más probable es que cuenten con tarjetas de red Ethernet.

2.2.3.1.1 10BASE5

Tiene un conector DB-15 hembra tipo D de 15 pines, el cable que se utiliza es el Thin Ethernet que se acopla al resto de la red a través de un transmisor-receptor externo.

2.2.3.1.2 10BASE2

Tiene un conector BNC que se une al cable por medio de un T BNC y las otras dos partes de la T están conectadas al cable Thinnet que va hacia otros nodos de la red. Aunque Thinnet tiene topología física de bus, en realidad el adaptador de red se bifurca y se dirige hacia el segmento de cable de red mediante el conector T BNC, por lo que si se retira el conector de la tarjeta no afectará la conexión de la red.

2.2.3.1.3 10BASE-T

Los adaptadores de red que soportan 10BASE-T usan un conector RJ-45, similar al conector RJ-11 de la mayor parte de las instalaciones telefónicas, a excepción de que el conector RJ-45 es más grande y tiene ocho conductores en vez de cuatro. El cable de red UTP tiene una clavija RJ-45 de la tarjeta de red y el otro extremo al socket RJ-45 del concentrador, los otros nodos de la red se conectan al concentrador en forma similar.

2.2.3.2 Dirección MAC

Las tarjetas de red tipo Ethernet tienen una pequeña memoria en la que alojan un dato único para cada tarjeta de este tipo. Se trata de la dirección MAC, y está formada por 48 bits que se suelen representar mediante dígitos hexadecimales que se agrupan en seis pares (cada par se separa de otro mediante dos puntos ":" o mediante guiones "-").

Por ejemplo, una dirección MAC podría ser F0:E1:D2:C3:B4:A5.

La mitad de los bits de la dirección MAC son usados para identificar al fabricante de la tarjeta, y los otros 24 bits son utilizados para diferenciar cada una de las tarjetas producidas por ese fabricante.

2.2.4 Cables para red

En este punto se debe decidir que tipo de cable se utilizará para conectar las terminales y el servidor. Debido a que la instalación y la elección del cable constituyen generalmente el cincuenta por ciento de la instalación completa, se deben planear cuidadosamente estos aspectos.

Cada estándar de red define el tipo de cableado que se requiere y las especificaciones para la conexión de los nodos de la red. A causa de las altas velocidades y a la gran cantidad de datos que se transmiten a través del cable, las especificaciones del mismo y las reglas para su uso son muy estrictas. El uso de cable con especificaciones incorrectas causa a final de cuentas, fallas en las comunicaciones de la red.

Los estándares especificados para la longitud del cable se refieren por lo general a segmentos de cable. Un segmento de cable es una sección continua que no es interrumpida por ningún dispositivo, como un concentrador o un repetidor. Se pueden incluir conectores en un segmento de cable de red, a causa de que sencillamente se conectan en el segmento sin interrumpir o cambiar la señal de la red.

Existen compañías especializadas en cables de datos y de telecomunicaciones que pueden configurar cables de longitud a la medida, de acuerdo con requerimientos particulares.

Si se tiene que instalar un vasto cableado o si se tienen requerimientos de cable especiales, se puede comprar el cable a granel e instalar uno mismo los conectores. Si se escoge este último método, es indispensable que se obtengan las herramientas y los conectores adecuados.

2.2.4.1 Ethernet

Hay cuatro tipos de cable para una red Ethernet: Thick Ethernet (10BASE5), Thin Ethernet (10BASE2), UTP (10BASE-T) y fibra óptica (10BASE-F). Los cables de red utilizados para cada tipo de Ethernet no pueden mezclarse, aunque existen dispositivos que permiten la conexión de diferentes tipos de segmentos de red Ethernet.

2.2.4.1.1 Thick Ethernet

El cable empleado por Thick Ethernet (Ethernet gruesa) es un tipo especial de cable coaxial. El conductor central está rodeado por un aislante dieléctrico al que a su vez, lo rodea un blindaje de hoja de metal. Alrededor del blindaje, hay un conductor tejido rodeado por otro blindaje de hoja de metal que, también está cubierto por un conductor tejido. La parte externa del cable tiene una cubierta protectora.

La reglas para la instalación y configuración de segmentos de cable Thick Ethernet son las siguientes:

- La longitud máxima de segmento de red es de 500 metros.
- Cada segmento de red debe tener una terminación de 50 ohms (ohmios) en cada extremo.
- No pueden conectarse en serie más de cinco segmentos de red y solo tres de éstos pueden estar ocupados (tener nodos conectados a ellos).
- La cantidad máxima de transmisores-receptores por segmento es de 100.
- La cantidad máxima de nodos en una red es de 1024.
- Los transceptores no pueden instalarse a menos de 2.5 metros.
- Los cables de bajada no pueden ser más largos de 50 metros.

A los conectores para el cable Thick Ethernet (Ethernet gruesa) se les llama conectores coaxiales serie N.

2.2.4.1.2 Thin Ethernet

El cable Thin Ethernet (Ethernet delgada), es un tipo de cable coaxial RG-58 que consiste en un conductor interno rodeado por un aislante dieléctrico, un blindaje de hoja de metal, un conductor tejido y una cubierta exterior protectora.

Para la instalación y configuración del cable Thin Ethernet, se aplican las siguientes reglas:

- La longitud máxima del segmento debe ser de 185 metros.
- Cada segmento de red debe tener una terminación de 50 ohms (ohmios) en cada extremo.
- No pueden conectarse en serie más de cinco segmentos de red y solo tres de éstos pueden estar ocupados (tener nodos conectados a ellos).
- La cantidad máxima de nodos por segmento es 30.
- La distancia mínima de cable entre adaptadores de red es 0.5 metros.
- La cantidad máxima de nodos en una red es de 1024.

Se usan conectores tipo BNC para el Thin Ethernet.

2.2.4.1.3 UTP Ethernet

El UTP es un cable que consiste en pares trenzados entre ellos, Ethernet UTP emplea un total de cuatro conductores (o dos pares) para transmitir y recibir la señal de red. Puesto que los conectores estándar RJ-45 tienen ocho números de conexión, el cable

que se instala tiene generalmente ocho conductores aunque la red solo use cuatro de ellos.

Adicionalmente al cableado de par trenzado viene en varios niveles que van desde el nivel 1 (o Categoría 1) utilizado con frecuencia para aplicaciones telefónicas, hasta nivel 5 (o Categoría 5) que está certificado para la transmisión de datos a una velocidad de hasta 155 megabits por segundo.

Para la instalación y configuración del cable Ethernet UTP, se aplican las siguientes reglas:

- La longitud máxima de cable entre un nodo y un concentrador es de 100 metros.
- La especificación IEEE para Ethernet 10Base-T requiere usar solo dos pares trenzados los pines 1, 2, 3 y 6 del conector RJ-45 son conectados de manera directa, donde 1 y 2 son transmisores y el 3 y 6 receptores.
- Se pueden conectar hasta 12 concentradores a un concentrador principal.
- Sin el uso de puentes, el cable Ethernet UTP puede acomodar un máximo de 1024 estaciones de trabajo.

Es importante conocer las normas para la configuración de los pines de un cable de red, ya que en algunos casos se necesita construir un cable directo y en otros un cable cruzado.

De acuerdo con la Norma EIA/TIA 568A RJ45, el orden de los pines es el siguiente:

Pin 1	Verde/Blanco
Pin 2	Verde
Pin 3	Naranja/Blanco
Pin 4	Azul
Pin 5	Azul/Blanco
Pin 6	Naranja
Pin 7	Café/Blanco
Pin 8	Café

Tabla 2-1 Norma EIA/TIA 568A RJ45

Y la Norma EIA/TIA 568B RJ45, especifica el siguiente orden:

Pin 1	Naranja/Blanco
Pin 2	Naranja
Pin 3	Verde/Blanco
Pin 4	Azul
Pin 5	Azul/Blanco
Pin 6	Verde
Pin 7	Café/Blanco
Pin 8	Café

Tabla 2-2 Norma EIA/TIA 568B RJ45

Para conectar una PC a un concentrador o a un switch se necesita un cable directo, este cable tendrá en ambos extremos la misma norma, pero si lo que se requiere es conectar 2 dispositivos del mismo tipo, esto se hace por medio de un cable cruzado; este cable se caracteriza porque en sus extremos se utilizan diferentes normas.

2.2.4.1.4 Fibra óptica

El cable de fibra óptica transmite datos por medio de una serie de pulsos de luz, transmitidos a través de una hebra fina de vidrio. Es capaz de transmitir datos a alta velocidad y no es susceptible de interferencias exteriores, la instalación requiere de cuidados especiales ya que sus empalmes y conectores son más costosos y difíciles de manejar. Actualmente se dispone de adaptadores de red que se conectan directamente a la fibra, ésta se emplea, por lo general, para conetar redes que requieren comunicación de alta velocidad en distancias mucho más largas.

2.2.5 Dispositivos de expansión

El consumidor promedio de una red de área local generalmente busca una solución rentable a sus problemas inmediatos, también desea incrementar la calidad y hacer más fácil su trabajo. Sin embargo, no todas las redes de área local fueron diseñadas teniendo como consideración principal el rendimiento.

Uno de los puntos que se deben tomar en cuenta al escoger una red, son la disposición física de la misma y los requerimientos de carga de trabajo previstos, de manera que se pueda determinar el posible rendimiento de la misma.

Si se necesita cubrir una distancia mayor se deberá escoger un hardware de red que lo permita. En el laboratorio de electrónica se cuenta con un switch como dispositivo de expansión, pero existen otros que pueden extender la longitud de la red. A continuación se describen los dispositivos usados por Ethernet.

Cada uno de los dispositivos y de los métodos usados para expandir la red tiene un propósito específico. Sin embargo muchos de ellos incorporan las características de otros tipos de dispositivos para aumentar su flexibilidad.

2.2.5.1 Repetidor

Un repetidor es un dispositivo que permite extender la longitud o cobertura de una red ya que amplifica y retransmite las señales, toma la señal que circula por un segmento de una red homogénea y la reenvía sin efectuar ningún tipo de interpretación; es capaz de conectar diferentes medios físicos de transmisión, sin embargo no suele utilizarse para conectar redes de banda base con redes de banda ancha ya que los métodos de decodificación de la información son muy diferentes.

2.2.5.2 Hub

Los hubs o concentradores son equipos que permiten compartir el uso de una línea o punto de red entre varias computadoras, todas ellas pueden usar la línea aunque no de forma simultánea, ni utilizando distintos protocolos, ni distintas velocidades de transmisión.

Los hubs funcionan repitiendo cada paquete de datos en cada uno de los puertos con los que cuenta de forma que todos los puntos tienen acceso a los datos.

Estos dispositivos son un punto central de conexión para nodos de red que están dispuestos de acuerdo a una topología física de estrella, como en el Ethernet 10BASE-T.

2.2.5.3 Switch

Los conmutadores (switch ethernet) se caracterizan por no enviar los paquetes a todos los puertos, sino únicamente al puerto correspondiente del destinatario. La diferencia entre un conmutador y un puente es que el puente debe recibir todo el paquete antes de dirigirlo al puerto correspondiente y un conmutador dirige el paquete a su destino una vez recibido el encabezado del paquete (en ella se encuentra la dirección IP del destinatario); gracias a ello, los conmutadores producen un retraso mínimo en la conmutación.

De esta manera, utilizando un conmutador se puede dividir una red en varios segmentos y limitar el tráfico al segmento o segmentos a los que pertenece el paquete. Su utilización permite que cada usuario o grupo de usuarios tenga su propio segmento y ancho de banda dedicados con mucha menor tasa de colisiones y un menor tiempo de respuesta en lugar de lo que ocurre en una red Ethernet tradicional en la que muchos usuarios compartan el mismo ancho de banda.

2.2.5.4 Puente

Un puente es un dispositivo que conecta dos redes LAN separadas para crear lo que aparenta ser una sola. Los puentes revisan la dirección asociada con cada paquete de información, si la dirección es la correspondiente al otro segmento de red, el puente pasará el paquete al segmento; si el puente reconoce que la dirección es la correspondiente a un nodo del segmento actual, no dejará pasar el paquete al otro lado.

Los puentes también suelen emplearse para reducir la cantidad de tráfico en una red, mediante la división de un solo segmento de red en dos y conectándolos por medio de éste.

2.2.5.5 Ruteador

Los ruteadores son similares a los puentes, sólo que operan a un nivel diferente, estos requieren por lo general que cada red tenga el mismo NOS. Con un NOS común, el ruteador puede ejecutar funciones más avanzadas de las que podría permitir un puente, como conectar redes basadas en topologías lógicas completamente diferentes como Ethernet y Token Ring. Los ruteadores también suelen ser lo suficientemente inteligentes para determinar la ruta más eficiente para el envío de datos, en caso de haber más de una ruta. Sin embargo, junto con la complejidad y la capacidad adicionales proporcionadas por los ruteadores se da una penalización de aumento de costo y un rendimiento disminuido (los ruteadores cuestan mucho más que los puentes).

2.2.5.6 Compuerta

Una compuerta (gateway) es un sistema formado por hardware y software que permite que los nodos de una red se comuniquen con tipos diferentes de red o con otros dispositivos. Podría tenerse, por ejemplo, una LAN que consista en computadoras compatibles con IBM y otra que consista en computadoras Macintosh. En este caso, una compuerta permitiría que las computadoras IBM compartieran archivos con las Macintosh. Este tipo de compuertas también permite que se compartan impresoras entre las dos redes.

2.3 Software de red

Para poder hacer una selección adecuada del NOS, es importante tener una opinión objetiva de personas que han experimentado, o están de alguna manera involucradas en el funcionamiento de estos sistemas, ya que al haberse enfrentado personalmente con uno o más de ellos, tienen una perspectiva un poco más amplia de sus pros y contras.

El software de red consiste en programas que implementan protocolos o normas de comunicación, para que las computadoras puedan compartir su información; éstos protocolos se aplican enviando y recibiendo grupos de datos formateados denominados paquetes.

Éste software también se encarga de efectuar conexiones lógicas entre las aplicaciones de la red, dirigir el movimiento de paquetes a través de la red física y minimizar las posibilidades de colisión entre paquetes enviados simultáneamente. En general este conjunto de programas forman parte de lo que se conoce como sistema operativo de red.

2.3.1 Network Operative System

El NOS permite que el software de aplicación (el procesador de palabras, las bases de datos, las hojas de cálculo, los paquetes de contabilidad, etc.) que se esté ejecutando en una computadora se comunique a través de la red con otras computadoras. El propósito principal de un sistema operativo es dar el soporte que necesitan los programas que hacen un trabajo de interés para el usuario. Por ejemplo, se puede estar utilizando un editor si se quiere crear un documento, ese editor no puede realizar su trabajo sin la ayuda del sistema operativo, necesitará ésta ayuda para interactuar con la terminal, los archivos y el resto de la computadora. Debido a que existen varios sistemas operativos de red en el mercado, es necesario observar la manera en que éstos están posicionados en el gusto del comprador.

Novell con sus diferentes versiones de NetWare fue el principal protagonista en el campo de las redes desde la década de los 80's. Microsoft tuvo un inicio lento con su sistema Windows NT, pero tuvo un gran auge que lo llevó a estar en segundo sitio detrás de Novell. Con menos fuerza se repartieron lo que restaba del mercado; el sistema OS/2 de IBM, la solución para redes cliente/servidor AppleShare de Apple y las diferentes variantes de Unix como Solaris, UnixWare y Linux.

Al momento de seleccionar el sistema operativo de red para una LAN, se necesita tener información sobre ciertos aspectos, que pueden influir de manera directa o indirecta en el éxito de la implementación del NOS en la red. Algunos de estos requerimientos son: personal capacitado, conocimiento de costos de cada NOS,

requerimientos de hardware y compatibilidad de los sistemas a instalar con otros similares.

Se propone utilizar Linux, que es un sistema operativo multiusuario y multitarea, tiene las mismas características que Unix pero también aporta elementos propios con la gran ventaja de ser software libre. Linux soporta el modo protegido de los procesadores de Intel 80x86, pero también las prestaciones más avanzadas de todos los procesadores que soporta su núcleo: PowerPC, SPARC, MIPS, sistemas IBM 390, PDA's, etc.

El entorno gráfico le aporta a Linux vistosidad por un lado pero facilidad de manejo por otro, al igual que los entornos gráficos de otros sistemas (Microsoft Windows y Apple Mac) X Window ofrece un entorno multiventana. En Linux se pueden ejecutar distintos gestores de ventanas, desde los conocidos Motif y Open Look originarios de Unix, hasta los más conocidos en la actualidad, Gnome y KDE.

Algunas características técnicas de este sistema operativo son:[9]

- Multitarea. Varios programas (realmente procesos) ejecutándose al mismo tiempo.
- Multiusuario. Varios usuarios en la misma máquina al mismo tiempo.
- Multiplataforma. Corre en muchas CPU's distintas, no sólo Intel.
- Tiene protección de la memoria entre procesos, de manera que uno de ellos no pueda detener el sistema.
- Memoria virtual usando paginación (swap).
- Todo el código fuente está disponible, incluyendo el núcleo completo y todos los drivers, las herramientas de desarrollo y todos los programas de usuario.
- Consolas virtuales múltiples. Varias sesiones de login a través de la consola entre las que se puede cambiar con las combinaciones adecuadas de teclas (totalmente independiente del hardware de video). Se crean dinámicamente y puedes tener hasta 64.
- Soporte para varios sistemas de archivo comunes, incluyendo minix-1, Xenix y todos los sistemas de archivo típicos de System V.
- TCP/IP, incluyendo FTP, telnet, NFS, etc.
- Software cliente y servidor Netware disponible en los núcleos de desarrollo.

2.3.2 Protocolos

Muchos componentes del NOS están basados en diversos protocolos, los cuales definen la forma en que se va a comunicar la computadora con otros dispositivos dentro de una red. Se puede definir un protocolo como el conjunto de normas que regulan la comunicación (establecimiento, mantenimiento y cancelación) entre los distintos componentes de una red informática. Existen dos tipos de protocolos: de bajo nivel y de red.

---

[9] FLORIANO Blanco Lino, GALÁN Galán Susana, Linux Al día en una hora, Ed. Anaya Multimedia

Los protocolos de bajo nivel controlan la forma en que las señales se transmiten por el cable o medio físico. Los protocolos de red organizan la información (controles y datos) para su transmisión por el medio físico a través de los protocolos de bajo nivel.

En Linux se tiene implementado el conjunto de protocolos de red TCP/IP, los cuales permiten comunicar e identificar a los diferentes dispositivos dentro de una red por medio de direcciones IP.

2.3.3 Dirección IP

El Protocolo de Internet (IP) y el Protocolo de Transmisión (TCP), fueron desarrollados inicialmente en 1973 por el informático estadounidense Vinton Cerf como parte de un proyecto dirigido por el ingeniero norteamericano Robert Kahn y patrocinado por la Agencia de Programas Avanzados de Investigación (ARPA, siglas en inglés) del Departamento Estadounidense de Defensa.

La especificación original de TCP/IP era abierta y así lo pensaron sus diseñadores, ellos crearon un espacio de direcciones o una manera estándar de escribirlas, la cual establecía 2^{32} direcciones (es decir, 4 294 967 296 direcciones diferentes). Estas direcciones permiten identificar de manera única a una computadora dentro de toda la red de Internet, por lo que se puede enviar información a otra computadora con solo saber su dirección IP; todo el proceso para llevar la información hasta la máquina destino queda en manos de la infraestructura propia de Internet (gateways, ruteadores, puentes, etc.).

Debido a que cada host en Internet requiere de un nombre y dirección únicos, una vez que se agote el espacio actual de direcciones IP de cuatro mil millones de direcciones, no habrá más direcciones. Esta es la razón por la cual es tan importante la siguiente generación del Protocolo de Internet llamada IPv6.

La razón por la cual las direcciones IP se han agotado tan rápido se debe al diseño del esquema de direccionamiento, todas las direcciones IP se escriben en notación decimal punteada, con un byte entre cada punto. Una dirección IP tienen el siguiente formato: X.X.X.X. Debido a que cada número está descrito por un byte, y que cada byte tiene 8 bits; cada número puede tener un valor cualquiera entre 0 y 225. Ya que existen 4 números con 8 bits cada uno, se dice que el espacio total de direcciones tiene una longitud de 32 bits.

Con un espacio de direcciones de 32 bits se podría pensar que nunca se terminarían las direcciones IP; sin embargo debido a la forma en que ha crecido Internet está sucediendo lo contrario, donde las direcciones IP se asignan a organizaciones que las solicitan en lo que se llaman bloques de direcciones. Los bloques de direcciones tienen tres tamaños, basándose en lo que se le conoce como clases de direcciones.

2.3.3.1 Clasificación

Existen 5 clases de direcciones de redes: A, B, C, D o E (esta diferenciación viene dada en función del número de computadoras que va a tener la red).

- Direcciones clase A: Contiene 7 bits para direcciones de red (el primer bit del primer octeto siempre es un 0) y los 24 bits restantes representan a direcciones de equipo. De esta manera, permite tener un máximo de 128 redes (aunque en realidad tiene 126, ya que están reservadas las redes cuya dirección de red empieza por 0 y por 127), cada una de las cuales puede tener 16, 777, 216 computadoras (aunque en realidad tiene 16, 777, 214 computadoras cada una, ya que se reservan aquellas direcciones de equipo en binario, cuyos valores sean todos ceros o todos unos). Las direcciones en representación decimal, estarán comprendidas entre 0.0.0.0 y 127.255.255.255 donde la máscara de subred será de 255.0.0.0.

- Direcciones clase B: Contiene 14 bits para direcciones de red (ya que el valor de los 2 primeros bits del primer octeto son siempre 10) y 16 bits para direcciones de equipo, lo que permite tener un máximo de 16, 384 redes, cada una de las cuales puede tener 65, 536 computadoras (aunque en realidad tiene 65, 534 computadoras cada una, ya que se reservan aquellas direcciones de equipo en binario cuyos valores sean todos ceros o todos unos). Las direcciones en representación decimal, estarán comprendidas entre 128.0.0.0 y 191.255.255.255 donde su máscara de subred será de 255.255.0.0.

- Direcciones clase C: Contiene 21 bits para direcciones de red (ya que el valor de los 3 primeros bits del primer octeto son siempre 110) y 8 bits para direcciones de equipo, lo que permite tener un máximo de 2, 097, 152 redes, cada una de las cuales puede tener 256 computadoras (aunque en realidad tiene 254 computadoras cada una, ya que se reservan aquellas direcciones de equipo en binario cuyos valores sean todos ceros o unos). Las direcciones en representación decimal, estarán comprendidas entre 192.0.0.0 y 223.255.255.255, por lo que su máscara de subred será de 255.255.255.0.

- Direcciones clase D: Se reserva todas las direcciones para multidestino (multicasting), es decir, una computadora transmite un mensaje a un grupo específico de computadoras de esta clase. El valor de los 4 primeros bits del primer octeto son siempre 1110 y los últimos 28 bits representan los grupos multidestino. Las direcciones en representación decimal estarán comprendidas entre 224.0.0.0 y 239.255.255.255.

- Direcciones clase E: Se utiliza con fines experimentales únicamente y no está disponible para el público. El valor de los 4 primeros bits del primer octeto son siempre 1111 y las direcciones en representación decimal estarán comprendidas entre 240.0.0.0 y 255.255.255.255.

2.3.3.2 Máscara de subred

La máscara de subred indica la localización de cada parte de una dirección IP, es decir, cuántos bits corresponden al identificador de la red y subred, además de cuántos se destinan al identificador de host.

Está formada por 32 bits que toman valor 1 ó 0

- Valor 1 para los bits cuya posición corresponde a bits de identificación de red
- Valor 0 para los bits cuya posición corresponde al identificador de host

Para que su manejo sea más sencillo se dividen en octetos, se pasan a decimal, y se separan por puntos. Los bits correspondientes a la identificación de subredes se toman siempre contiguos para obtener un rango continuo de direcciones de host. La configuración de cualquier dispositivo en TCP/IP requiere siempre una dirección IP única y el valor correspondiente a su máscara de subred ya sea genérica o personalizada para crear subredes.

Para cada clase de direcciones hay definidas mascaras genéricas:

- CLASE A: 255.0.0.0
- CLASE B: 255.255.0.0
- CLASE C: 255.255.255.0

A la hora de decidir que máscara de subred se utilizará hay que tener en cuenta la red y su posible crecimiento, es decir el aumento en número de subredes y el aumento en número de equipos. El LTSP particularmente utiliza la máscara de subred de la clase C "255.255.255.0", la cual permitirá tener hasta 254 terminales y un servidor.

2.3.4 Arquitectura cliente - servidor

La arquitectura cliente/servidor es un modelo para el desarrollo de sistemas de información, en el que las transacciones se dividen en procesos independientes que cooperan entre sí para intercambiar información, servicios o recursos. Se denomina cliente al proceso que inicia el diálogo o solicita los recursos, y servidor al proceso que responde a las solicitudes. Es el modelo de interacción más común entre aplicaciones en una red, no forma parte de los conceptos de Internet como los protocolos IP, TCP o UDP, sin embargo todos los servicios estándares de alto nivel propuestos en Internet funcionan según este modelo. Los principales componentes del esquema cliente/servidor son los clientes, los servidores y la infraestructura de comunicaciones.

En este modelo, las aplicaciones se dividen de forma que el servidor contiene la parte que debe ser compartida por varios usuarios, y en el cliente permanece sólo lo particular de cada usuario. Los clientes interactúan con el usuario, usualmente en forma

gráfica, frecuentemente se comunican con procesos auxiliares que se encargan de establecer conexión con el servidor, enviar el pedido, recibir la respuesta, manejar las fallas y realizar actividades de sincronización y de seguridad. Los servidores proporcionan un servicio al cliente y devuelven los resultados.

En algunos casos existen procesos auxiliares que se encargan de recibir las solicitudes del cliente, verificar la protección, activar un proceso servidor para satisfacer el pedido, recibir su respuesta y enviarla al cliente. Además, deben manejar los interbloqueos, la recuperación ante fallas, y otros aspectos afines. Por las razones anteriores, la plataforma computacional asociada con los servidores es más poderosa que la de los clientes.

Para que los clientes y los servidores puedan comunicarse se requiere una infraestructura de comunicaciones, la cual proporciona los mecanismos básicos de direccionamiento y transporte. La mayoría de los sistemas cliente/servidor actuales, se basan en redes locales y por lo tanto utilizan protocolos no orientados a conexión, lo cual implica que las aplicaciones deben hacer las verificaciones. La red debe tener características adecuadas de desempeño, confiabilidad, transparencia y administración.

Como ejemplos de clientes pueden citarse interfaces de usuario para enviar comandos a un servidor, APIs para el desarrollo de aplicaciones distribuidas, herramientas en el cliente para hacer acceso a servidores remotos (por ejemplo, servidores de SQL) o aplicaciones que solicitan acceso a servidores para algunos servicios. Como ejemplos de servidores pueden citarse servidores de ventanas X Window, servidores de archivos como NFS, servidores para el manejo de bases de datos (como los servidores de SQL), servidores de diseño y manufactura asistidos por computadora, etc.

La red de terminales gráficas que se implementará en el laboratorio de electrónica, utilizará la arquitectura cliente-servidor, ya que contará con un servidor que proveerá los servicios o programas a los clientes que en este caso serán las terminales gráficas; las cuales usarán estos recursos para poder realizar alguna tarea de interés para el usuario. A su vez en un determinado momento, las terminales recibirán peticiones por parte del servidor para que cada una de ellas puedan desplegar en pantalla los resultados de los procesos.

2.4 Aspectos básicos en la instalación de Fedora Core Linux

Para garantizar el buen desempeño del servidor donde será implementado el LTSP y facilitar la instalación de Fedora Core, es necesario conocer detalladamente los principales aspectos que intervienen durante y posteriormente a la instalación de éste sistema operativo.

2.4.1 Tipos de instalación

Fedora Core proporciona cinco opciones de instalación, las cuales definen una serie de paquetes a instalar; es decir, dependiendo del uso que se le vaya a dar a la computadora y de los programas que se piensen utilizar, se instalará el software que más se adecue a las necesidades del usuario.

2.4.1.1 Escritorio personal

Es apropiado para estaciones de trabajo o portátiles, se debe seleccionar este tipo de instalación para crear un entorno gráfico y un sistema ideal para el uso del directorio principal o del escritorio; además es la más adecuada para principiantes.

2.4.1.2 Estación de trabajo

Esta opción instala un entorno gráfico con herramientas para el desarrollo del software y la administración del sistema.

2.4.1.3 Servidor

Este tipo de instalación es útil si se desea compartir archivos, impresoras, así como los diferentes servicios web. Se habilitarán también los servicios adicionales y se puede escoger si se desea o no instalar un entorno gráfico.

2.4.1.4 Personalizada

Con esta opción se tiene el control completo sobre el proceso de instalación, se podrá elegir el esquema de particionamiento, incluyendo la selección de los paquetes de software y las preferencias de la autenticación.

2.4.1.5 Actualización

Ésta es la solución más adecuada si ya se tiene instalada una versión de Fedora Core, y se quiere actualizar rápidamente a los últimos paquetes y versiones del kernel. A partir de las necesidades específicas de cada usuario, éste podrá elegir el tipo de instalación que más le conviene.

2.4.2 Métodos de instalación

Para instalar algún programa adicional en Linux, se deben efectuar ciertos pasos utilizando las herramientas apropiadas. Algunas distribuciones manejan sus propios métodos, pero los más conocidos y los que serán empleados en este trabajo serán los siguientes.

2.4.2.1 Código fuente (TGZ)

Las aplicaciones GNU son distribuidas en distintos formatos para realizar su instalación, entre ellos se encuentra su código fuente, no solo por el hecho de que se debe cumplir con la filosofía del software libre; si no también para permitir al usuario optimizar la aplicación a instalar para las características de su computadora.

Esta optimización se logra al compilar directamente el código fuente en la máquina donde se va a ejecutar el programa, ya que se toma en cuenta la arquitectura especifica de la misma, como el procesador, la memoria, etc. Para garantizar una compilación exitosa, es recomendable leer previamente los archivos de ayuda INSTALL y README, sin embargo por lo general los pasos para compilar el código fuente de una aplicación son los siguientes:

- Copiar el archivo .tgz, al directorio donde será descomprimido y compilado. Por ejemplo al directorio de root.

```
cp  archivo.tgz  /root
```

- Descomprimir el archivo: Para archivos con extensión .tgz o .tar.gz se utiliza el siguiente comando:

```
tar -zxvf  nombre_archivo
```

Para archivos con extensión .bz2 o .tar.bz2

```
tar -lxvf  nombre_archivo
```

- Se habrá creado un directorio con el nombre de la aplicación, se debe entrar a él con el comando:

```
cd nombre_directorio
```

- Se debe configurar el código fuente para determinar si se puede compilar y que lo hará sin errores. El archivo de configuración "configure" se utiliza para comprobar las dependencias de las bibliotecas necesarias; para garantizar que se cuente con ellas, es aconsejable que durante la instalación de Linux se seleccionen todas las herramientas de desarrollo, que incluyen generalmente el código fuente del software X, del Kernel, además de los entornos de escritorio GNOME y KDE.

  Se debe teclear lo siguiente:

```
./configure
```

  Esto creará los archivos "makefiles" necesarios para la compilación.

- Para iniciar la compilación solo se teclea lo siguiente:

```
make
```

Al final de la compilación se habrán generado el o los ejecutables optimizados para el sistema.

- Finalmente se instalan los ejecutables generados:

```
make install
```

2.4.2.2 RPM

El Administrador de paquetes Red Hat Package Manager (RPM) es un sistema de empaquetado de software que se utiliza en Fedora Core además de otras distribuciones Linux y UNIX. Los paquetes RPM se caracterizan por tener la extensión .rpm, el nombre del archivo incluye el nombre del paquete, su versión y la arquitectura para la cual fue optimizado.

Un RPM es un "paquete" o archivo que contiene (comprimido) un programa ejecutable, sus archivos de ayuda, los programas pre-instalados o dependencias que necesita para poder funcionar, algunos archivos de configuración y una serie de scripts para automatizar su instalación y desinstalación. RPM facilita la instalación y actualización del software, ya que es posible hacerlo por medio de comandos breves. Este sistema mantiene una base de datos de los paquetes y de sus archivos respectivos, por lo que se pueden realizar consultas y verificaciones de los paquetes ya instalados.

Los comandos básicos para el manejo de paquetes RPM, son los siguientes:

- Instalación:

```
rpm –ivh nombre_archivo.rpm
```

- Desinstalación:

```
rpm –e nombre_paquete_instalado
```

- Búsqueda de paquetes instalados:

```
rpm –qa | grep palabra_clave
```

2.4.3 Particionamiento

Las particiones garantizan que la capacidad de disco necesaria para determinados tipos de procesos no va a ser utilizada por otros. Esto ayuda a asegurar que los sistemas y procesos sigan ejecutándose, incluso si se saturan algunas particiones. Fedora Core proporciona 3 opciones de particionamiento:

- Particionamiento automático: Se configuran las particiones basándose en el tipo de instalación, espacio disponible en el disco duro, tamaño de la RAM y crea por default las siguientes particiones: Swap, Raiz (/), /boot. También se pueden personalizar las particiones restantes y adecuarlas si es necesario.

- Disk Druid: La herramienta del particionamiento manual del disco, permite configurar las particiones en un entorno interactivo. Se pueden configurar los tipos de sistemas de archivos, puntos de montaje, tamaño y otras opciones.

- Fdisk: Esta herramienta de particionamiento, es una utilidad de modo texto recomendada tan solo a usuarios avanzados que necesitan ejecutar tareas especializadas.

2.4.3.1 Particiones básicas en un sistema Linux

Todo sistema Linux debe tener un mínimo de 2 particiones, una para el sistema de archivos y por lo menos una partición de intercambio de memoria (swap). Se recomienda que al menos haya 2 particiones del sistema de archivos: una para los directorios que tiendan a crecer y la otra para los directorios que tengan que mantener espacio libre para que el sistema funcione. Para determinar el número y tamaño de las particiones en un sistema Linux, no hay alguna regla rigurosa; sin embargo, en algunos

textos se recomiendan configuraciones tanto para una estación de trabajo como para un servidor. [10]

- Particiones para una estación de trabajo

/	Partición root	500 MB – 1000 MB
/home	Archivos de usuario	1000 MB en adelante
swap	Memoria de intercambio	doble de RAM

- Particiones para un servidor

/boot	Partición boot (kernel/LILO)	2 MB – 100 MB
/	Partición root	80 MB – 100 MB
/usr	Partición de sistema	500 MB – 1000 MB
/usr/local	Partición para aplicaciones	500 MB en adelante
/opt	Partición para aplicaciones	500 MB en adelante
/home	Archivos de usuario	1000 MB en adelante
swap	Memoria de intercambio	Doble de RAM

Otra configuración propuesta es la siguiente:[11]

- Particiones para una estación de trabajo

/boot	Partición boot (kernel/LILO)	2 MB – 100 MB
/	Partición root	500 MB en adelante
swap	Memoria de intercambio	Doble de RAM

- Particiones para un servidor

/boot	Partición boot (kernel/LILO)	16 MB
/	Partición root	300 MB
/usr	Partición de sistema	300 MB – 700 MB
/home	Archivos de usuario	512 MB en adelante
/tmp	Datos temporales	200 MB en adelante
swap	Memoria de intercambio	Doble de RAM

En la práctica, si no se cuenta con suficiente espacio en el disco duro se puede reducir el tamaño de las particiones; tratando de que la disminución sea proporcional para todas. Incluso es posible omitir algunas, y Linux se encargará de gestionar el uso de las particiones existentes, pero esto incrementa el riesgo de que en algún momento si se corrompe alguna partición, se pierdan datos que originalmente deberían estar almacenados en otro lugar del disco.

Hasta este punto hemos conocido los diferentes componentes de red así como aspectos que son necesarios para la implementación del proyecto, ahora podremos instalar los programas que deben estar presentes en el servidor previamente a los paquetes del LTSP, así como su correcta configuración.

---

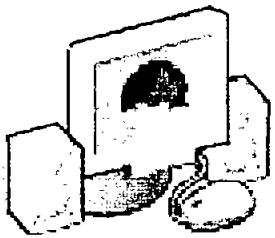
[10] KOPLER Michael, WESLEY Addison, Linux Installation, Configuration and Use, Ed. Great Britain, Great Britain, 2000, p.38

[11] PRITCHARD Kara J., Red Hat Certified Exam, Ed. Coriolis, USA, 2000, p.10

IMPLEMENTACIÓN

OBJETIVO ESPECÍFICO

Conocer las características y consideraciones generales para la instalación y configuración de los diferentes componentes tanto de hardware como de software, que intervienen en la implementación del LTSP.



3

INTRODUCCIÓN

En este capítulo se profundizará en los conocimientos técnicos específicos, así como las herramientas que serán necesarias para entender el funcionamiento, implementación y mantenimiento del LTSP.

Es importante definir lo que se necesita para comenzar con la implementación del proyecto, una parte fundamental es la elección de la máquina servidor, ésta debe contar con un buen procesador, suficiente memoria y un disco duro de capacidad aceptable. Principalmente se prevé todo esto para un mejor rendimiento, tomando en cuenta los servicios que va a ofrecer y a cuantas terminales podrá atender eficientemente.

Se enfatiza la forma de cómo redimensionar el disco duro para no perder la información que se tiene en Windows, debido a que se permitirá que el servidor preste servicios de estación de trabajo si así se requiere. Hay que recordar que una de las características importantes de Linux es que es capaz de leer y escribir en sistemas de archivos FAT, VFAT, FAT32, además de que encontramos infinidad de aplicaciones de libre distribución que se encuentran disponibles.

Se mostrará paso a paso cómo instalar Linux especificando las opciones que son requeridas para el ambiente LTSP, así como las que mejor se adapten a las necesidades propias del laboratorio.

Finalmente se instalará y configurará el LTSP, proporcionando las condiciones adecuadas para la siguiente fase del proyecto.

3.1 Requerimientos

El LTSP es un proyecto que requiere de ciertos componentes, los cuales deben cumplir con especificaciones que permitirán su adecuado funcionamiento y rendimiento al trabajar conjuntamente, para ello es necesario definir algunos requerimientos que permitan dar una referencia de lo que realmente se necesita para su implementación. A continuación se mencionarán los requerimientos recomendados de hardware y software.

3.1.1 Hardware

Una red de terminales gráficas debe contar con un servidor que cumpla con ciertas especificaciones, para que éste pueda atender de forma eficiente las peticiones que se le hagan. Las terminales demandan recursos del servidor para poder funcionar y por lo tanto su rendimiento depende de las capacidades que el servidor pueda ofrecer.

Existen otros factores que también determinan los requerimientos físicos del servidor, como son: el tipo de programas o servicios que se van a ejecutar, el número de usuarios que van a utilizar las terminales, la arquitectura de la red, entre otros.

Tomando en cuenta estos factores, se mencionarán los requerimientos de hardware recomendados para el servidor. A pesar de que con otras configuraciones se pueden obtener buenos resultados, incluso con un servidor de menor capacidad; es importante considerar que un servidor de mejores características implica un mayor rendimiento en las terminales.

Para un servidor con 5 terminales, los requerimientos recomendados son:[12]

- CPU Intel PIII, 1 GHz
- 512 MB RAM
- Disco duro 20 GB, IDE ATA 100/133
- (2) tarjetas de red Ethernet base 10/100
- 1 CD-ROM
- 1 Floppy

Para las terminales los requerimientos recomendados son:

- CPU 486 DX, 40MHz
- 16 MB RAM
- Tarjeta de red Ethernet base 10
- Tarjeta de video VGA
- 1 Floppy

[12] Pág. web del proyecto K12LTSP, Sección de Instalación http://k12ltsp.org/install.html

Para implementar un servidor LTSP se requiere tener al menos 256 MB de RAM, más 50 MB por cada terminal; algunas aplicaciones, por ejemplo las que manipulan imágenes, necesitarán más memoria por lo que se recomienda tener 1 GB.

Si la red tiene más de seis terminales, es recomendable contar con discos SCSI; estos discos se caracterizan por que son mucho más rápidos que los IDE, además de que permiten accesos múltiples a los mismos.

La velocidad del procesador también depende del uso que se le dará a los equipos; por ejemplo las aplicaciones de ofimática requieren de una velocidad relativamente pequeña (1 GHz). Al igual que con la memoria, las aplicaciones gráficas necesitan una mayor velocidad del procesador.

En cuanto a las tarjetas de red, las de 100 Mb funcionan bastante bien para una red que tenga hasta 30 terminales; además, es necesario contar con un Switch o un Hub, que permita implementar el modelo cliente-servidor. Se recomienda usar un Switch, ya que tiene control sobre colisiones y esto disminuye el tráfico en la red.

En el laboratorio de electrónica hay computadoras que satisfacen algunas de estas características; por lo que no será necesario adquirir algún equipo nuevo, además la idea de implementar este proyecto es precisamente el de seguir utilizando las computadoras con las que ya se cuenta, aunque no se descarta la posibilidad de que en un futuro se puedan mejorar las características del servidor.

Las características de este equipo son las siguientes:

Equipo 1

- CPU Intel 486, 66 MHz
- 16 MB RAM
- Disco duro 512 MB, IDE ATA 100
- Tarjeta de red 3Com 509 EtherLink III, bus ISA

Equipo 2

- CPU Intel 486, 66 MHz
- 16 MB RAM
- Disco duro 512 MB, IDE ATA 100
- Tarjeta de red 3Com 509 EtherLink III, bus ISA

Equipo 3

- CPU Intel PIII, 700 MHz
- 128 MB RAM
- Disco duro 20 GB IDE ATA 100
- Tarjeta de red MX98715, bus PCI

Equipo 4

- CPU Intel PIII, 700 MHz
- 384 MB RAM
- Disco duro 20 GB IDE ATA 100
- Tarjeta de red Sis 900, bus PCI
- Tarjeta de red RTL8139C, bus PCI

Se optó por utilizar el último equipo como servidor debido que cuenta con las mejores características.

3.1.2 Software

La computadora que será utilizada como servidor ya tiene instalado el sistema operativo Windows, sin embargo éste no reúne las características para implementar el LTSP; por esta razón es necesario instalar otro sistema operativo, para este caso Linux ya que cuenta con capacidades tanto de estación de trabajo como de servidor.

3.1.2.1 Redimensionar la partición

Debido a que en el laboratorio se trabaja con aplicaciones que solo pueden correr bajo Windows se optó por conservarlo, éste ocupa toda la capacidad del disco duro, es decir; solo existe una partición tipo FAT 32 sin embargo Linux requiere sus propias particiones.

Para liberar espacio en el disco duro donde estarán las particiones de Linux, se requiere modificar el tamaño de la partición de Windows, para lo cual se hará uso de la herramienta llamada Partition Magic que se ejecuta desde MS-DOS o Windows y permite crear, borrar o redimensionar las particiones de un disco de manera sencilla y confiable.

A continuación se muestra el proceso para liberar el espacio en disco para Linux:

Después de ejecutar el Partition Magic se muestra la partición de Windows que ocupa los 20 GB totales, pero se necesita redimensionarla para que solo sea de 10 GB, el espacio restante será asignado a las particiones de Linux.

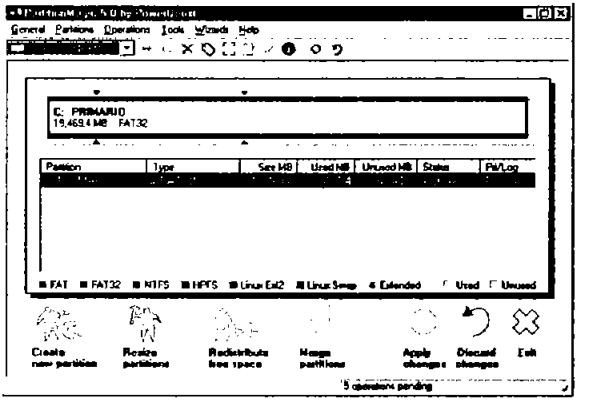


Figura 3-1 Se muestra la partición de Windows que ocupa 20 GB

Posteriormente se elige resize/move del menú operations, que muestra una ventana donde se pueden modificar las propiedades de la partición.

El tamaño de la partición se puede cambiar de 2 maneras, la primera posicionándose al final de la barra y arrastrando el puntero del ratón hasta fijar el nuevo tamaño y la segunda introduciendo el valor correspondiente en MB en el campo New Size, por último se da click en el botón OK para aceptar los cambios.

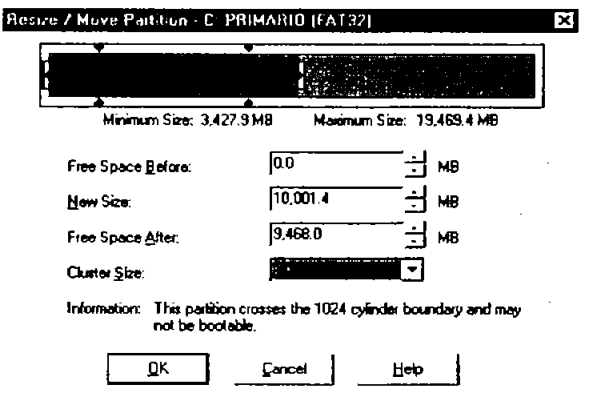


Figura 3-2 Proceso de liberación de espacio en disco para Linux

Ahora se puede observar el nuevo tamaño de la partición de Windows y el espacio libre para Linux.

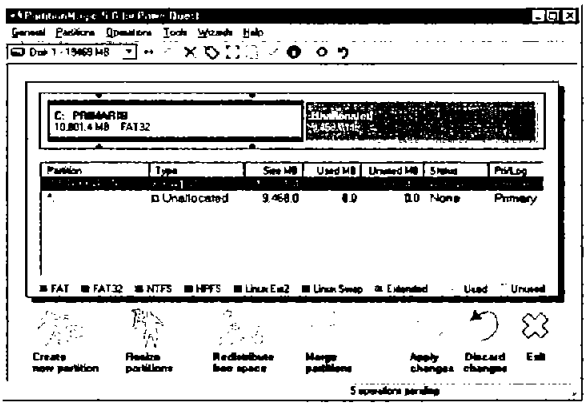


Figura 3-3 Nuevo tamaño de la partición de Windows y el espacio para Linux

Finalmente se da click en Apply Changes para efectuar los cambios, el programa tendrá que cerrar Windows para comenzar con el proceso de redimensionamiento. Una vez que haya finalizado este proceso, se podrá comenzar con la instalación de Linux. Si no se desea conservar la instalación de Windows, se puede formatear el disco duro, para que Linux ocupe todo el espacio disponible.

3.1.2.2 Instalación del sistema operativo

En esta sección se describen las opciones y configuraciones recomendadas durante la instalación de Fedora Core. En este caso el método de instalación será vía CD-ROM, por lo que es necesario configurar el BIOS de la computadora para que permita bootear desde este dispositivo.

Paso 1: Se introduce el CD 1 y automáticamente se inicia el programa de instalación de Fedora Core llamado Anaconda, en caso de que el programa no pueda ser iniciado se tiene que crear un disco de arranque, éste permitirá cargar una imagen del sistema operativo con la información necesaria para poder realizar la instalación por medio del CD-ROM.

El disco de arranque se crea de la siguiente manera:

Desde MS-DOS, se utiliza el comando rawrite, el cual viene incluido en el primer disco de instalación de Fedora Core, en la carpeta DOSUTILS.

Se escribe el comando:

```
D:\DOSUTILS\rawrite
```

Se pedirá la ruta de la imagen de arranque:

```
D:\IMAGES\boot.img
```

Y después la ruta donde se escribirá la imagen de arranque, en este caso es el disquete:

```
A:\
```

Se tiene que reiniciar la computadora para que ésta arranque por medio del disco generado.

Paso 2: El programa de instalación muestra una pantalla de inicio, donde no se pide información alguna. La sección de ayuda aparece por defecto, es recomendable leer la información mostrada.

Para continuar con la instalación, se debe hacer click en el botón "Siguiente".

Paso 3: Seleccionar el idioma español, éste es el lenguaje que será utilizado para la instalación y para el sistema. Hacer click en "Siguiente" para continuar.

Paso 4: Seleccionar el tipo de teclado apropiado para el sistema, en este caso se elige "español". Hacer click en "Siguiente" para continuar.

Paso 5: Se debe seleccionar el tipo de ratón, en la mayoría de los casos será detectado automáticamente. Si el ratón es de 2 botones, se puede habilitar la opción para emular 3 botones. Hacer click en "Siguiente" para continuar con el proceso.

Paso 6: Debido a la flexibilidad que se requiere para la instalación de un servidor de terminales gráficas, se debe seleccionar el tipo de instalación más apropiado; en este caso será la instalación "personalizada". Para continuar, se debe hacer click en "Siguiente".

Paso 7: En este punto se tiene que elegir una de las opciones de particionamiento del disco duro. Seleccionar "Disk Druid" y hacer click en "Siguiente".

Paso 8: Se debe tener cuidado al crear las particiones ya que de esto depende el buen funcionamiento del sistema.

Disk Druid muestra gráficamente las particiones del disco duro, en este caso la partición de Windows y el espacio libre destinado para Linux.

La primer partición que hay que crear es la swap, para esto se debe seleccionar con el ratón el espacio libre del disco duro, y hacer click en el botón "nuevo" lo cual abre un cuadro de diálogo donde se puede configurar las características de la nueva partición.

Se debe seleccionar en el campo "Tipo de sistema de archivos" la opción "Swap", posteriormente se especifica su tamaño, el cual será de 768 MB. Finalmente se habilita la opción "tamaño fijo". Hacer click en Aceptar para cerrar el cuadro de diálogo.

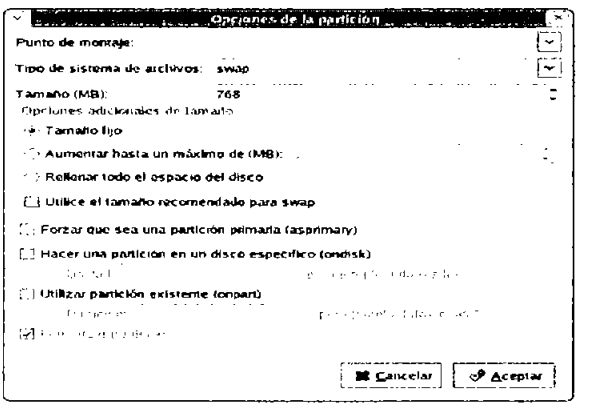


Figura 3-4 Proceso de creación de la partición swap

La segunda partición que debe ser creada es la /boot, que se utiliza para almacenar los archivos de arranque del sistema.

Con ayuda del ratón se debe seleccionar el espacio libre del disco duro, y hacer click en el botón "nuevo" lo cual abre el cuadro de diálogo para configurar la nueva partición.

En la sección punto de montaje se debe elegir "/boot" y como sistema de archivos el tipo "ext3". Esta partición tendrá un tamaño de 100 MB, por lo que se debe habilitar la opción "tamaño fijo".

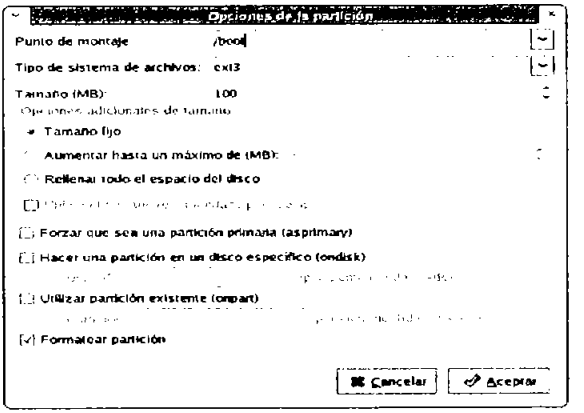


Figura 3-5 Proceso de creación de la partición boot

La última partición que se tiene que crear es la raíz (/), para esto debe seleccionarse el espacio libre del disco duro, posteriormente se hace click en el botón "nuevo" para abrir el cuadro de diálogo correspondiente.

En la sección "punto de montaje" se elige "/" y como sistema de archivos el tipo "ext3", en la sección "opciones de tamaño" se debe seleccionar "completar hasta el tamaño máximo permitido"; lo cual permite seleccionar todo el espacio libre del disco duro.

Hacer click en Aceptar para cerrar el cuadro de diálogo.

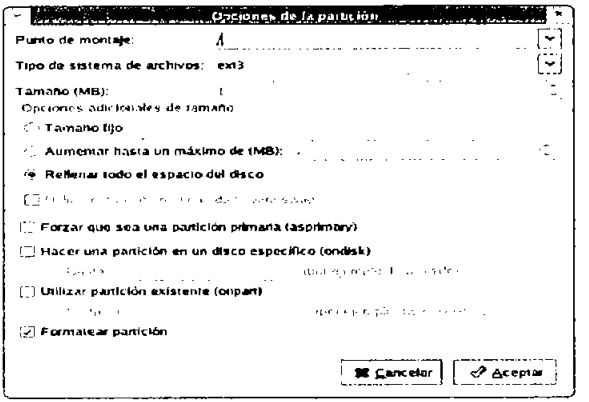


Figura 3-6 Proceso de creación de la partición raíz

Para finalizar el proceso de creación de las particiones de debe hacer click en "siguiente".

Paso 9: El programa Anaconda, instala por defecto el gestor de arranque GRUB, el cual permitirá elegir con que sistema operativo iniciar (Windows o Fedora Core). Hacer click en "Siguiente" para continuar.

Paso 10: Fedora Core permite proteger el sistema mediante un firewall, para instalar un nivel aceptable de seguridad, se debe elegir la opción "nivel intermedio". Hacer click en "Siguiente" para continuar.

Paso 11: Es necesario instalar un idioma para usarlo por defecto, éste será utilizado por el sistema una vez que la instalación se haya completado. En este caso se debe seleccionar "Español (México)". Hacer click en "Siguiente" para continuar.

Paso 12: El programa de instalación permite seleccionar un huso horario, el cual será "América / Ciudad de México". Hacer click en "Siguiente".

Paso 13: En este paso se tiene que proporcionar la cuenta root, la cual es usada para instalar paquetes, actualizar RPM's y realizar la mayoría de las tareas de mantenimiento del sistema. Una vez que se haya ingresado y confirmado la contraseña, se debe hacer click en "Siguiente".

Paso 14: En esta parte es donde se eligen los paquetes que serán instalados en el sistema.

A continuación se muestra una lista con los grupos de paquetes a instalar.

- Escritorios
  - Sistema X Window
  - GNOME
  - KDE

- Aplicaciones
  - Editores
  - Ingeniería y científico
  - Internet gráfica
  - Oficina/productividad
  - Sonido y video
  - Gráficos

- Servidores
  - Servidores de red (elegir DHCP)

- Desarrollo
  - Herramientas de desarrollo
  - Desarrollo del kernel

- Desarrollo del software X
- Desarrollo de software de GNOME
- Desarrollo de software para KDE

➢ Sistema
- Herramientas de administración
- Soporte para la impresión

Paso 15: El programa de instalación formateará las particiones e iniciará la instalación de los paquetes necesarios. Durante esta etapa, se le solicitará el ingreso de los CD's 2 y 3.

Paso 16: Una vez finalizado el proceso de instalación de los paquetes, se puede crear un disco de arranque; este disco se utiliza en caso de que el gestor de arranque GRUP no funcione. Hacer click en continuar.

Paso 17: Una vez que se hayan completado todos los pasos anteriores, el programa finalizará la instalación.

3.2 Instalación y configuración de los paquetes en el servidor.

Hasta ahora se conoce de manera general cual es la forma en que trabaja el LTSP, sin embargo es necesario comprender a fondo todo lo que implica la implementación de este proyecto; para ello se describirá la función y la instalación de cada uno de los programas que deben estar presentes en el servidor previamente a los paquetes del LTSP. Posteriormente se mostrará la forma en que deben ser configurados para que funcionen en conjunto con el LTSP.

Se da por hecho que el servidor ya debe estar corriendo el sistema operativo Fedora Core, y que se cuenta con los permisos de root.

3.2.1 DHCP

Dynamic Host Configuration Protocol (Protocolo de Configuración Dinámica de Servidor) es un protocolo de red que se utiliza para asignar automáticamente información TCP/IP a equipos cliente. Cada uno de ellos se conecta a un servidor DHCP centralizado que devuelve la configuración de red del cliente, incluida la dirección IP, el gateway y los servidores DNS.

DHCP es útil para proporcionar de un modo rápido la configuración de red de una máquina cliente, al configurar el sistema cliente el administrador puede seleccionar

el protocolo DHCP y no especificar una dirección IP, una máscara de red, un gateway o servidor DNS fijos; el cliente recupera esta información desde el servidor.

DHCP también es útil si un administrador desea cambiar la dirección de IP de muchos sistemas; en lugar de volver a configurarlos todos, puede modificar un archivo de configuración en el servidor para establecer la nueva dirección IP.

Si los servidores DNS de una organización cambian, los cambios también se aplicarán en el servidor DHCP, no en todos los clientes. Una vez que se reinicie la red, se aplicarán los cambios.

Además, si una computadora portátil o cualquier tipo de equipo móvil se configura para hacer peticiones DHCP, podrá desplazarse entre distintas oficinas sin tener que volver a configurarse, ya que cada oficina dispondrá de un servidor DHCP que permitirá su conexión a la red.

3.2.1.1 Instalación de los paquetes RPM

Para la instalación del servicio DHCP, se debe descargar la última versión de los paquetes y ejecutar los siguientes comandos:

```
rpm -ivh dhcp-devel-3.0pl1-9.i386.rpm
rpm -ivh dhcp-3.0pl1-9.i386.rpm
```

3.2.2 TFTP

Trivial File Transfer Protocol (Protocolo Simple de Transferencia de Archivos) es un protocolo que utiliza el UDP (Protocolo de Datagramas de Usuario) para transferir archivos entre distintas máquinas conectadas a través de una red.

Se implementa sobre un servicio de comunicaciones no fiable y no orientado a conexión, por lo que no requiere el nombre de un usuario ni su contraseña, lo que da lugar a que carezca de seguridad. TFTP ejecuta las mismas tareas que el FTP (Protocolo de Transferencia de Archivos), pero usa un protocolo de transporte diferente.

3.2.2.1 Instalación de los paquetes RPM

Para la instalación del servicio TFTP, se debe descargar la última versión de los paquetes y ejecutar los siguientes comandos:

```
rpm -ivh tftp-0.29-3.i386.rpm
rpm -ivh tftp-server-0.29-3.i386.rpm
```

3.2.3 NFS

Network File System (Sistema de Archivos de Red) es un método para compartir archivos entre máquinas de una red, de tal forma que parece que se está trabajando en un disco duro local. Esto permite centralizar archivos en una localización, mientras se permite su acceso continuo a los usuarios autorizados.

Fedora Core puede trabajar como servidor o como cliente de NFS (o ambos), lo que implica que puede exportar sistemas de archivos a otras máquinas, así como montar los sistemas de archivos que otras computadoras exportan.

NFS resulta útil para compartir directorios de archivos entre múltiples usuarios de la misma red. Por ejemplo, un grupo de usuarios que trabajan en un mismo proyecto pueden tener acceso a los archivos del mismo usando una porción compartida del sistema de archivos NFS, que se ha montado en un directorio determinado, como pudiera ser /myproject. Para acceder a los archivos compartidos; el usuario accede al directorio /myproject de su máquina local.

Linux usa una combinación de soporte a nivel de kernel y demonios en continua ejecución para proporcionar la disponibilidad de los archivos vía NFS, y el servicio debe estar activo en el kernel de Linux para que funcione. NFS usa Remote Procedure Calls (Llamadas de Procedimientos Remotos) para enrutar peticiones entre clientes y servidores, implicando que el servicio portmap deba estar disponible y activo en los niveles de ejecución adecuados para que la comunicación funcione.

Con NFS, la autentificación solo se produce cuando el cliente intenta montar un sistema de archivos remoto. Para limitar el acceso, el servidor NFS utiliza los archivos /etc/hosts.allow y /etc/hosts.deny para determinar si a un cliente en particular le debe ser explícitamente permitido o denegado su acceso.

Después de revisar los archivos /etc/hosts.allow y /etc/hosts.deny, el servidor NFS recurre a su archivo de configuración, /etc/exports, para determinar si el cliente tiene suficientes privilegios para montar alguno de los sistemas de archivos exportados. Los privilegios de montaje NFS son permitidos específicamente a clientes, no a usuarios. Si se permite a una máquina acceder a un sistema de archivos exportado, los usuarios de esa máquina podrán acceder a esos datos.

3.2.3.1 Instalación de los paquetes RPM

Para la instalación del servicio NFS, se debe descargar la última versión de los paquetes y ejecutar los siguientes comandos:

```
rpm -ivh nfs-utils-1.0.1-2.i386.rpm
rpm -ivh portmap-4.0-46.i386.rpm
```

3.2.4 LTSP

Existen una serie de paquetes para instalar el entorno LTSP, hay 3 grupos principales los cuales son: Core, Linux Kernel y X Windows. Se dividieron los paquetes en grupos, para mejorar su funcionalidad y flexibilidad.

3.2.4.1 Paquete Core

Este paquete proporciona los scripts de instalación del LTSP, los cuales permiten crear la estructura de directorios que será compartida a cada una de las terminales y también los archivos plantilla para la configuración de los servicios requeridos.

Se deberá conseguir la última versión del paquete y ejecutar el siguiente comando:

```
rpm –ivh ltsp_core-3.0.7-4.i386.rpm
```

3.2.4.2 Paquete Ltsp_Kernel

Este paquete contiene un kernel con un ramdisk (disco virtual). Dentro del ramdisk se incluye soporte para todos los módulos de las tarjetas de red y un espacio de usuario con un cliente dhcp llamado dhclient.

Se deberá conseguir la última versión del paquete y ejecutar el siguiente comando:

```
rpm –ivh ltsp_kernel-3.0-5.i386.rpm
```

3.2.4.3 Paquete X Windows

Este paquete es necesario si se desea configurar una terminal gráfica (Interfaz gráfica GUI), el cual proporciona todos los drivers pertenecientes al XFree86; es recomendable que también se instale el paquete de fuentes X Fonts. Este paquete viene separado del X Core debido a que en la mayoría de los casos se utiliza un servidor de fuentes X (Xfs).

Se deberá conseguir la última versión de los paquetes y ejecutar los siguientes comandos:

```
rpm –ivh ltsp_x_core-3.0.4-2.i386.rpm
rpm –ivh ltsp_x_fonts-3.0.4-2.i386.rpm
```

3.2.4.4 Scripts de configuración LTSP

Después de instalar los paquetes principales se requiere ejecutar algunos scripts que son necesarios para configurar los archivos de sistema del servidor. Cada uno de estos scripts es responsable de modificar o en su caso crear archivos de ejemplo de los servicios NFS, TFTP, DHCP, etc.

Se accede al directorio install_scripts con el siguiente comando:

```
cd /opt/ltsp/install_scripts
```

Posteriormente se ejecuta el siguiente script para crear el directorio templates.

```
sh ./install.sh
```

Este script muestra la siguiente salida:

```
About to install LTSP, using the following settings:

 LTSP_DIR    = /opt/ltsp
 SWAP_DIR    = /var/opt/ltsp/swapfiles
 TFTP_DIR    = /tftpboot
 IP_NETWORK  = 192.168.0.0
 IP_SERVER   = 192.168.0.254
 IP_NETMASK  = 255.255.255.0
 IP_BROADCAST = 192.168.0.255

If you want to install LTSP using the above settings,
enter 'Y' and the installation will proceed.  Any other
response will abort the installation, and you can modify
the CONFIG file and restart the installation.

Continue with installation (y/n)?

find: i386: No existe el archivo o el directorio
./install.sh: line 274: /opt/ltsp/i386/etc/version: No existe el archivo o el directorio

Take a look in /tmp/ltsp.install.log for a complete log of the installation

You now need to change to the /opt/ltsp/templates directory and
run the ltsp_initialize script to complete the installation

./install.sh: line 397: /opt/ltsp/i386/etc/lts.orig: No existe el archivo o el directorio
```

Tabla 3-1 Creación del directorio templates

Aunque se muestran ciertos errores, estos pueden ignorarse; lo importante en este punto es que se haya creado el directorio templates.

Una vez creado el directorio se tendrá que entrar a él.

```
cd ../templates
```

Estando en el directorio templates se ejecuta lo siguiente:

```
./ltsp_initialize
```

Este comando preguntará qué servicios se quieren configurar, para ello tendrá que crear o modificar ciertos archivos. La salida que muestra este comando es la siguiente:

```
Linux Terminal Server Project (http://www.LTSP.org)

The following files will be created/modified:

/etc/X11/xdm/Xaccess              The config file to allow remote xdm log  [Y]
/etc/X11/xdm/Xsetup_workstation   Sets the logo of your login window       [Y]
/etc/dhcpd.conf.example           Example config file for dhcp             [Y]
/etc/exports                      The config file for nfs                  [Y]
/etc/X11/gdm/gdm.conf             The config file for gdm                  [Y]
/etc/X11/gdm/Init/Default         The gdm startup script                   [Y]
/etc/hosts.allow                  Configuration file for tcp wrappers      [Y]
/etc/inittab                      Config file for init                     [Y]
/etc/kde/kdm/kdmrc                The config file for kdm                  [Y]
/etc/X11/xdm/ltsp.gif             The background logo for your login       [Y]
/etc/rc.d/rc5.d/S60nfs            Startup links for nfs                    [Y]
/etc/rc.d/rc5.d/S13portmap        Startup links for portmapper             [Y]
/etc/sysconfig/syslog             Startup info for syslogd                 [Y]
/etc/xinetd.d/tftp                Enable the tftp daemon                   [Y]
/etc/X11/xdm/xdm-config           The main config file for xdm/kdm         [Y]


Ready to apply the changes? ( R-Review, A-Apply, C-Cancel )
```

Tabla 3-2 Ejecución de scrip ltsp.initialize en el directorio templates

Debido a que en ocasiones algunos de estos archivos son configurados manualmente por el administrador por cuestiones de seguridad, es recomendable verificar las modificaciones que realiza este script. En caso de estar convencidos de que no se va a comprometer la seguridad del sistema, se pueden aplicar los cambios; lo que generará o en su caso modificará los archivos mencionados.

3.2.5 Configuración de los archivos de sistema

Los scripts del LTSP, crearán entradas (líneas de código) en los archivos de configuración de los diferentes servicios necesarios; estas entradas son configuraciones por default de lo que se necesita para que el servidor LTSP funcione adecuadamente. Algunos parámetros podrán modificarse o adaptarse dependiendo de la configuración de la red, como por ejemplo las direcciones IP; sin embargo es recomendable conservar los valores iniciales. En algunos casos se crearán líneas comentadas, las cuales servirán de ejemplo en caso de requerir configuraciones especiales, esto sucede a menudo debido a la gran diversidad del hardware disponible. A continuación se muestran los archivos de sistema más importantes para la implementación del LTSP.

3.2.5.1 dhcpd.conf

Este archivo de configuración consta de un conjunto de sentencias, las cuales se clasifican en parámetros y declaraciones. Los parámetros expresan como hacer algo, si se hace algo o no, así como los atributos que se le asignan al cliente. Las declaraciones, en cambio, se emplean para describir la topología de una red, describir a un conjunto de clientes o para aplicar determinados parámetros a un grupo de declaraciones.

Las declaraciones tienen la forma:

```
<nombre de la declaración> [atributos] {
        [parámetros]
        [declaraciones]
}
```

y los parámetros:

```
[option] <nombre del parámetro> [valores];
```

Los parámetros que comienzan con la palabra reservada "option" describen aquellos datos que brinda el servidor al cliente como parte del protocolo, y los que no, describen las características del servidor de DHCP.

A continuación se describen las sentencias declarativas:

- shared-network: Permite agrupar un conjunto de subredes que compartan la misma red física. El único atributo de esta sentencia es un nombre que sólo se utiliza para las trazas del servicio.

Sintaxis:

```
shared-network <nombre> {
        [parámetros]
        [declaraciones]
}
```

- subnet: Permite agrupar las características globales que van a tener los clientes de una misma subred.

Sintaxis:

```
subnet <dirección de red>netmask<máscara de red> {
        [parámetros]
        [declaraciones]
}
```

- range: Permite definir un rango de direcciones IP a otorgar a clientes pertenecientes a una subred. Toda declaración tipo subnet debe tener asociada una declaración range en la cual se especifique las direcciones IP mínima y máxima. Si se especifica el atributo dynamic-bootp se indica que estas direcciones se pueden asignar también a clientes BOOTP. Cuando se especifica una sola dirección IP se omite la dirección máxima.

Sintaxis:

```
range [dynamic-bootp] <dirección IP mínima> [dirección IP máxima]
```

- host: Permite describir aquellos hosts que tengan una dirección fija. Todos los clientes que usan BOOTP deben tener asociada una sentencia host. Un cliente se corresponde con una declaración host si la opción dhcp-client-identifier indicada en la declaración posee el valor del identificador que brinda el cliente a través del protocolo. De no ser así entonces se emplearía la dirección MAC del cliente especificada a través del atributo hardware.

Sintaxis:

```
host <hostname> {
        [parámetros]
        [declaraciones]
}
```

- group: Permite agrupar a otras declaraciones para aplicarles varios parámetros comunes. Puede ser utilizada para agrupar hosts, subredes, redes compartidas y otros grupos.

Sintaxis:

```
group {
        [parámetros]
        [declaraciones]
}
```

Los principales parámetros son:

- lease-file-name <filename>: Indica el nombre del archivo donde se almacenan los contratos. Este parámetro tiene alcance global por lo que se debe especificar fuera de todos los ámbitos (declaraciones) para que tenga efecto real. Por defecto es /var/lib/dhcp/dhcpd.leases

- default-lease-time <time>: Expresa en segundos la duración de una conexión.

- max-lease-time <time>: Expresa en segundos la duración máxima de una conexión.

- min-lease-time <time>: Expresa en segundos la duración mínima de una conexión.

- min-seconds <seconds> : Indica el número de segundos que debe esperar el servidor DHCP para responder a la petición de los clientes. Se utiliza cuando se tiene un segundo servidor y se desea que este responda después que el otro haya atendido la petición del cliente.

- hardware <type> <address>: Indica la dirección física (MAC) de un cliente particular (declaraciones tipo host). El atributo type expresa el tipo de arquitectura de la interfaz de red, actualmente puede ser: ethernet o token-ring.

- server-name <servername>: Indica el nombre que se ofrecerá a los clientes como identificador del servidor que emplean.

- fixed-address <address> [, <address>]: Expresa las direcciones IP que son fijas para los clientes descritos a través de las declaraciones de tipo host. Pueden utilizarse nombres de dominio en lugar de números IP.

- dynamic-bootp-lease-cutoff <date>: Indica la fecha en que expiran los contratos de todos los clientes BOOTP.

- get-lease-hostnames <flag>: Indica si el servidor resolverá o no las direcciones IP de los clientes a nombres de dominio y usará estos nombres como la opción host-name del protocolo.

- use-host-decl-names <flag>: Indica si se asume que el nombre provisto en cada una de las declaraciones tipo host dentro del mismo ámbito, es el nombre del cliente correspondiente (opción host-name del protocolo).

- authoritative;
- non authoritative;

Indican si el servidor está autorizado o no para realizar sus funciones. Por defecto un servidor DHCP asume que la información que brinda a una subred determinada no es correcta ni tampoco está autorizado para brindar el servicio.

Esto permite que si un usuario inexperto instala un servidor de DHCP en la red este no sea escuchado por los clientes como lo es un servidor legítimo que se le indique explícitamente que está autorizado.

El administrador de red que configure adecuadamente su servidor debe colocar este parámetro al comienzo del archivo, aunque puede ser conveniente en algunas ocasiones declarar al servidor autorizado de acuerdo a los segmentos de red definidos y no de forma global.

- always-broadcast <flag>: Se emplea para algunos clientes de DHCP/BOOTP que no pueden recibir las respuestas del servidor si no son en forma de broadcast. Se debe tratar de colocar este parámetro a "on" sólo para los clientes que realmente lo necesiten pues provoca demasiado tráfico en la red.

- ddns-update <flag>: Indica si se realizan o no actualizaciones dinámicas al DNS siempre que se establezca un contrato. Por defecto este parámetro tiene valor "on".

- allow <request>;
- deny <request>;
- ignore <request>;

Se emplean para controlar la respuesta del servidor DHCP ante distintos tipos de peticiones, algunas de las posibles solicitudes (atributo request) son:

  - unknown-clients: se emplea para indicar al servidor si acepta o no las solicitudes de los clientes desconocidos. Un cliente desconocido es aquel que no tiene asociado una declaración tipo host. Por defecto las solicitudes de estos clientes se aceptan.

- bootp: se utiliza para señalar si se aceptarán o no los pedidos de los clientes BOOTP. Por defecto se aceptan.

- booting: se emplea en las declaraciones del tipo host para indicar si se aceptará o negará la solicitud del host correspondiente. Por defecto se aceptan para todos los hosts.

- declines: se utiliza para indicar si el servidor acepta o no los mensajes del tipo DHCPDECLINE de los clientes. Cuando un servidor recibe este tipo de mensajes asume que la dirección que ofrece no es válida pues al parecer alguien no autorizado la está utilizando y entonces la declara como abandonada.

  Desafortunadamente un cliente "malicioso" o con una implementación incorrecta puede agotar todo el spool de direcciones a otorgar que posee el servidor y antes de que este decida emplear las direcciones abandonadas ya se habrán provocado algunos trastornos en el servicio.

Entre los parámetros que se le pueden otorgar a un cliente a través del protocolo y que van precedidos por la palabra "option", se encuentran:

- option domain-name <domain name>: Indica el nombre del dominio que empleará el cliente.

- option domain-name-servers <ip address> [, <ip address> ...]: Indica los servidores de nombres de dominio a emplear por el cliente.

- option host-name <hostname>: Indica el nombre que empleará el host cliente.

- option subnet-mask <ip address>: Indica la máscara de red que se le asignará al cliente.

- option routers <ip address> [, <ip address> ...]: Indica las direcciones IP de los routers (gateway) que empleará el cliente.

- option broadcast-address <ip address>: Indica la dirección de broadcast que utilizará el cliente.

- option dhcp-client-identifier <string>: Indica el identificador que puede emplear el cliente como alternativa a su dirección MAC.

Un ejemplo del archivo dhcpd.conf es creado como parte de la instalación del LTSP. Este archivo es llamado /etc/dhcpd.conf.example y puede ser copiado o renombrado a /etc/dhcpd.conf.

El archivo tiene la siguiente estructura:

```
# Sample configuration file for dhcpd
#
# Make changes to this file and copy it to /etc/dhcpd.conf
#
ddns-update-style       none;

default-lease-time          21600;
max-lease-time              21600;

option subnet-mask          255.255.255.0;
option broadcast-address    192.168.0.255;
option routers              192.168.0.254;
option domain-name-servers  192.168.0.254;
option domain-name          "yourdomain.com";
option root-path            "192.168.0.254:/opt/ltsp/i386";

option option-128 code 128 = string;
option option-129 code 129 = text;

shared-network WORKSTATIONS {
  subnet 192.168.0.0 netmask 255.255.255.0 {
  }
}

group   {
  use-host-decl-names     on;
  option log-servers      192.168.0.254;

  host ws001 {
    hardware ethernet   00:E0:06:E8:00:84;
    fixed-address       192.168.0.1;
    filename            "/lts/vmlinuz-2.4.19-ltsp-1";
##      option option-128   e4:45:74:68:00:00;  #This is NOT a MAC address
##      option option-129   "NIC=ne IO=0x300";
  }
  host ws002 {
    hardware ethernet   00:D0:09:30:6A:1C;
    fixed-address       192.168.0.2;
    filename            "/lts/vmlinuz-2.4.9-ltsp-6";
  }
}
```

Tabla 3-3 Ejemplo de configuración del archivo dhcpd

Una vez renombrado el archivo y hechas las modificaciones pertinentes el dhcpd puede ser iniciado con el siguiente comando:

```
service dhcpd start
```

3.2.5.1.1 Terminal 1

Para este equipo, la configuración del archivo dhcpd.conf debe ser:

```
host ws001 {
        hardware ethernet    00:60:8C:CA:BC:FF;
        fixed-address        192.168.0.1;
        filename             "/lts/vmlinuz-2.4.19-ltsp-1";
        option option-128    e4:45:74:68:00:00;
        option option-129    "NIC=3c509";
}
```

3.2.5.1.2 Terminal 2

Para este equipo, la configuración del archivo dhcpd.conf debe ser:

```
host ws002 {
        hardware ethernet    00:A0:24:24:F6:0F;
        fixed-address        192.168.0.2;
        filename             "/lts/vmlinuz-2.4.19-ltsp-1";
        option option-128    e4:45:74:68:00:00;
        option option-129    "NIC=3c509";
}
```

3.2.5.1.3 Terminal 3

Para este equipo, la configuración del archivo dhcpd.conf debe ser:

```
host ws003 {
        hardware ethernet    00:80:AD:3B:8F:E4;
        fixed-address        192.168.0.3;
        filename             "/lts/vmlinuz-2.4.19-ltsp-1";
}
```

3.2.5.2 tftp

El archivo tiene la siguiente estructura:

```
# default: off
# description: The tftp server serves files using the trivial file transfer\
#       protocol.  The tftp protocol is often used to boot diskless \
#       workstations, download configuration files to network-aware printers, \
#       and to start the installation process for some operating systems.
service tftp
{
        socket_type             = dgram
        protocol                = udp
        wait                    = yes
        user                    = root
        server                  = /usr/sbin/in.tftpd
        server_args             = -s /tftpboot
        disable                 = yes
        per_source              = 11
        cps                     = 100 2
        flags                   = IPv4
}
```

Tabla 3-4 Ejemplo de configuración del archivo tftp

En Fedora Core, el servicio tftp es iniciado por xinetd; el archivo de configuración tftp contiene la información necesaria para que corra tftpd. Por lo que el servicio podrá ser iniciado con el siguiente comando:

```
service xinetd start    ó
service xinetd restart
```

3.2.5.3 hosts

Por lo general las computadoras se comunican con direcciones IP, pero para las personas es más sencillo recordar nombres que números, es entonces cuando el DNS o el archivo /etc/hosts entra escena. Este mapeo de direcciones IP a nombres de host no es requerido generalmente, pero en un ambiente LTSP si lo es. Esto es así porque sin este mapeo, el NFS dará errores de permisos cuando la terminal intente montar el sistema de archivos raíz. Aparte de los problemas de NFS, si la terminal no se encuentra en el archivo /etc/hosts, también se podrían tener problemas con los Display Managers GDM y KDM.

El principal propósito de este archivo es resolver los nombres de hosts a partir de su dirección IP. Se puede usar solamente para resolver nombres de hosts en pequeñas redes sin servidor DNS.

Sin tener en cuenta el tipo de red que la computadora utilice, este archivo contiene una línea que especifica la dirección IP del dispositivo loopback (127.0.0.1) como por ejemplo localhost.localdomain; éste dispositivo hace referencia a la misma computadora.

La sintaxis de este archivo es muy simple:

```
<dirección IP> minombre.midominio.org alias
```

Se deben utilizar direcciones IP que pertenezcan al rango de direcciones IP privadas. Los rangos válidos son:

Clases de redes

A 10.0.0.0
B Entre 172.16.0.0 y 172.31.0.0
C Entre 192.168.0.0 y 192.168.255.0

El archivo tiene la siguiente estructura:

```
# Do not remove the following line, or various programs
# that require network functionality will fail.
127.0.0.1       localhost.localdomain       localhost
192.168.0.254   localhost.localdomain       localhost
```

Tabla 3-5 Ejemplo de configuración del archivo hosts

3.2.5.3.1 Terminal 1

Para este equipo, la configuración del archivo hosts debe ser:

```
192.168.0.1     ws001     ws001.localdomain     ws001
```

3.2.5.3.2 Terminal 2

Para este equipo, la configuración del archivo hosts debe ser:

```
192.168.0.2     ws002     ws002.localdomain     ws002
```

3.2.5.3.3 Terminal 3

Para este equipo, la configuración del archivo hosts debe ser:

```
192.168.0.3     ws003     ws003.localdomain     ws003
```

3.2.5.4 Hosts.allow, hosts.deny

En un servidor Linux el acceso a los servicios que usan wrappers TCP y que se basan en el nombre de la máquina, lo controlan dos archivos: /etc/hosts.allow y /etc/hosts.deny. Estos archivos usan un formato simple para controlar el acceso a los servicios.

Siempre y cuando no se haya especificado alguna regla en hosts.allow y hosts.deny, la regla por defecto consiste en permitir a cualquiera el acceso a los servicios.

El orden es importante ya que las reglas que se encuentran en hosts.allow tienen precedencia con respecto a las especificadas en hosts.deny. Aunque una regla deniegue el acceso a un determinado servicio en hosts.deny, los clientes que tengan el permiso para acceder al servicio en hosts.allow lo mantienen.

Las reglas de cada uno de los archivos están colocadas por orden de importancia por lo que hay que respetar dicho orden. Todos los cambios que se hagan en estos archivos afectan inmediatamente a los servicios a los que se refieran; no se requiere volver a arrancar los servicios. Todas las reglas del control de acceso que se encuentran en las líneas hosts.allow y hosts.deny, y los espacios en blanco o las líneas que empiezan con (#) no se toman en cuenta.

Cada regla tiene que estar en una línea separada.

Sintaxis:

```
<daemon_list>: <client_list>[: <shell_command> ]
```

Cada una de estas opciones se refiere a una parte específica de la regla:

- daemon_list: Es una colección de uno o varios nombres de procesos o de comodines especiales separados por espacios en blanco.

- client_list: Uno o varios nombres de máquinas, direcciones de máquina, funciones o comodines separados por espacios en blanco que se usan cuando el nombre de un determinado proceso se adapta a un determinado servicio.

- shell_command: Es un componente opcional que indica la realización de algo en el caso de que se use una regla.

Las características son de bastante ayuda cuando se especifican qué grupos de clientes pueden o no tener acceso a un determinado servicio. Así al poner el punto "." al comienzo de una cadena, todas las máquinas que se encuentran en esa cadena deben cumplir dicha regla. De tal manera que .domain.com engloba tanto a system1.domain.com como a system2.domain.com.

El punto "." al final de una cadena tiene el mismo efecto, excepto que el sentido cambia, ya que ésto se usa sobre todo para las direcciones IP ya que una regla relativa a 192.168.0. se aplica a todo el bloque C de las direcciones IP. Las expresiones de las máscaras de red también se pueden usar como característica para controlar el acceso a un determinado grupo de direcciones IP.

También puede usar asteriscos (*) o interrogaciones (?) para seleccionar grupos enteros de nombres de máquinas o de direcciones IP siempre y cuando no las use en la misma cadena de las otras características.

Si la lista de los nombres de máquinas que tienen acceso a un determinado servicio es muy larga o es difícil de controlar en los archivos host.allow o hosts.deny, puede especificar el recorrido completo a un archivo (como /etc/telnet.hosts.deny). Este archivo contiene diferentes nombres de máquinas, direcciones de máquinas o características separados por espacios en blanco a los que desea conceder o denegar el acceso a dicho servicio.

El método también funciona para compartir las listas de control del acceso entre varios servicios ya que los cambios que se quieran realizar solamente se tienen que hacer en un archivo para que afecten a cada servicio.

Los siguientes comodines se usan en las reglas para el control del acceso en vez de usar determinados nombres de máquinas o grupos de nombres de máquinas:

- ALL: Para permitir a un cliente tener acceso a todos los servicios, se utiliza ALL en la sección de los demonios.

- LOCAL: Hace corresponder todos los nombres de máquinas que no contengan un punto ".".

- KNOWN: Hace corresponder todas las máquinas cuyos nombres y direcciones se conozcan o en el caso en que se conozca el usuario.

- UNKNOWN: Hace corresponder todas las máquinas cuyos nombres y direcciones sean desconocidas o en el caso en el que se desconozca el usuario.

- PARANOID: Hace corresponder todas las máquinas cuyo nombre no se corresponda con la dirección.

El script de instalación del LTSP agregará algunas entradas en el archivo /etc/hosts.allow para permitir que dhcp, tftp y portmap funcionen correctamente. Estas entradas dan por hecho que se están usando direcciones IP de la clase C 192.168.0.0.

Si se utilizan direcciones diferentes se deben sustituir adecuadamente. Como el archivo hosts.allow tiene prioridad sobre el archivo hosts.deny, en realidad los cambios más importantes se efectúan en el primero.

El archivo hosts.allow tiene la siguiente estructura:

```
#
# hosts.allow    This file describes the names of the hosts which are
#                allowed to use the local INET services, as decided
#                by the '/usr/sbin/tcpd' server.
#

## LTS-begin ##

#
# The lines between the 'LTS-begin' and the 'LTS-end' were added
# on: sab sep 27 12:34:50 CDT 2003 by the ltsp installation script.
# For more information, visit the ltsp homepage
# at http://www.ltsp.org
#

bootpd:   0.0.0.0
in.tftpd: 192.168.0.
portmap:  192.168.0.

## LTS-end ##
```

Tabla 3-6 Ejemplo de configuración del archivo hosts.allow

Y el archivo hosts.deny, la siguiente:

```
#
# hosts.deny    This file describes the names of the hosts which are
#               *not* allowed to use the local INET services, as decided
#               by the '/usr/sbin/tcpd' server.
#
# The portmap line is redundant, but it is left to remind you that
# the new secure portmap uses hosts.deny and hosts.allow.  In particular
# you should know that NFS uses portmap!
```

Tabla 3-7 Ejemplo de configuración del archivo lts.conf

Estos archivos, permiten o niegan respectivamente algún servicio a las direcciones especificadas (en este caso las terminales), en algunas ocasiones pueden definirse direcciones IP concretas y en otros rangos de direcciones; incluso pueden combinarse.

Pero hay que tener cuidado al manipular estos archivos, ya que forman parte del firewall del sistema.

3.2.5.4.1 Terminal 1

Para este equipo, la configuración del archivo hosts.allow debe ser:

```
ALL:    192.168.0.1
```

En el archivo hosts.deny no es necesario hacer alguna configuración para esta terminal.

3.2.5.4.2 Terminal 2

Para este equipo, la configuración del archivo hosts.allow debe ser:

```
ALL:    192.168.0.2
```

En el archivo hosts.deny no es necesario hacer alguna configuración para esta terminal.

3.2.5.4.3 Terminal 3

Para este equipo, la configuración del archivo hosts.allow debe ser:

```
ALL:    192.168.0.3
```

En el archivo hosts.deny no es necesario hacer alguna configuración para esta terminal.

3.2.5.5 Exports

Es sencillo configurar un sistema para compartir archivos y directorios usando NFS, cada sistema de archivos que se exporta a usuarios remotos vía NFS, así como los derechos de acceso relativos a ellos, es localizado en el archivo /etc/exports. Este archivo es leído por el comando exportfs que da a rpc.mountd y rpc.nfsd la información necesaria para permitir el montaje remoto de un sistema de archivos por una máquina autorizada. Por omisión, rpc.mountd desaprueba el montaje de todos los directorios, lo cual es una medida de seguridad muy adecuada. Si se desea permitir montar un directorio de NFS, debe ser exportado, para lograr esto se debe especificar en el archivo exports.

Cada línea define un directorio y los anfitriones a los que se les permite montarlo. Un nombre de anfitrión es usualmente un nombre de dominio pero puede contener adicionalmente los comodines "*" y "?" los cuales funcionan igual que en el Bourne Shell de Linux. Por ejemplo, lab*.foo.com coincide con lab01.foo.com así como laboratory.foo.com. El anfitrión puede ser especificado también usando un rango de direcciones IP en la forma dirección/máscara de red.

Cuando se verifica a un anfitrión cliente contra el archivo exports, rpc.mountd busca el nombre del anfitrión cliente usando la llamada gethostbyaddr. Con DNS, esta llamada devuelve el nombre canónico del anfitrión cliente, así debe asegurarse de no usar alias en exports. En un entorno NIS el nombre devuelto es la primera coincidencia de la base de datos de anfitriones, y sin DNS o NIS, el nombre devuelto es el primer nombre de anfitrión encontrado en el archivo hosts que coincida con la dirección del cliente.

El nombre del anfitrión es seguido por una lista opcional de parámetros, los cuales pueden ser:

- secure: Asegura que la conexión se haga desde un puerto origen reservado, por ejemplo, uno que sea menor que 1024. Este parámetro es puesto por omisión.

- insecure: Este parámetro revierte el efecto de secure.

- ro: Provoca que el montaje de NFS sea para sólo lectura. Este parámetro está activado por omisión.

- rw: Esta opción monta la jerarquía de archivos en lectura-escritura.

- root_squash: Esta característica de seguridad deniega a los superusuarios en los hosts especificados cualquier derecho de acceso especial mapeando las peticiones desde el uid 0 en el cliente al uid 65534 (es decir, -2) en el servidor. Este uid debe ser asociado con el usuario nobody.

- no_root_squash: Esta opción no mapea las peticiones desde uid 0. Este parámetro está habilitado por omisión, así los superusuarios tienen acceso de supervisor a los directorios exportados de su sistema.

- link_relative: Convierte los enlaces simbólicos absolutos (donde el contenido del enlace comienza con un slash) en enlaces relativos. Esta opción sólo tiene sentido cuando está montado el sistema de archivos entero de un anfitrión; por otra parte, algunos de los enlaces podrían apuntar a ninguna parte, o peor aún, a archivos que nunca debieran apuntar. Esta opción está habilitada de forma predeterminada.

- link_absolute: Esta opción deja todos los enlaces simbólicos existentes.

- map_identity: Esta opción le indica al servidor asumir que el cliente usa el mismo uid y gid que el servidor. Este parámetro está habilitado por omisión.

- map_daemon: Indica al servidor de NFS asumir que el cliente y el servidor no comparten el mismo espacio uid/gid, rpc.nfsd entonces construye una lista que mapea los IDs entre cliente y servidor preguntando al demonio rpc.ugidd del cliente.

- map_static: Esta opción le permite especificar el nombre de un archivo que contiene un mapa estático de uids. Por ejemplo, map_static=/etc/nfs/vlight.map especificaría el archivo /etc/nfs/vlight.map como un mapa de uid/gid.

- map_nis: Esta opción causa que el servidor de NIS haga un mapeado de uid y gid.

- anonuid y anongid: Estas opciones le permiten especificar el uid y el gid de la cuenta anónima. Esto es útil si tiene un volumen exportado para montajes públicos.

En esencia, éste archivo define que directorios serán compartidos a otras máquinas, en el caso del LTSP, definirá los directorios a los cuales podrán tener acceso las terminales gráficas.

La estructura del archivo creada por los scripts del LTSP es la siguiente:

```
## LTS-begin ##
#
# The lines between the 'LTS-begin' and the 'LTS-end' were added
# on: Sat sep 27 12:34:50 CDT 2003 by the ltsp installation script.
# For more information, visit the ltsp homepage
# at http://www.ltsp.org
#

/opt/ltsp/i386              192.168.0.0/255.255.255.0(ro,no_root_squash,sync)
/var/opt/ltsp/swapfiles     192.168.0.0/255.255.255.0(rw,no_root_squash,async)

#
# The following entries need to be uncommented if you want
# Local App support in ltsp
#
#/home              192.168.0.0/255.255.255.0(rw,no_root_squash,sync)

## LTS-end ##
```

Tabla 3-8 Ejemplo de configuración del archivo exports

3.2.5.6 lts.conf

Este archivo contiene las configuraciones necesarias para crear el entorno LTSP; permite especificar los parámetros que las terminales requieren para poder funcionar correctamente. Este archivo tiene una sintaxis simple, que consiste de múltiples secciones. Hay una sección por defecto, llamada [default] y hay secciones adicionales para terminales individuales. Si todas las terminales son idénticas, entonces se deben especificar todas las opciones bajo la sección [Default]. Las terminales pueden ser identificadas por nombre de host, dirección IP o dirección MAC. A continuación se describen los parámetros principales que acepta el archivo lts.conf:

A) Opciones generales

- Comentarios: Los comentarios comienzan con un símbolo de número ('#') y continúan hasta el final de la línea.

- LTSP_BASEDIR: Indica dónde está ubicado el sistema de archivos raíz del LTSP; por defecto es en /opt/ltsp/.

- SERVER: Este es el servidor utilizado por XDM_SERVER, TELNET_HOST, XFS_SERVER y SYSLOG_HOST, si alguno de los anteriores no está especificado explícitamente. Si se tiene una computadora que está actuando como el servidor de

todo, entonces se puede especificar su dirección IP aquí y omitir el resto. Si ningún valor es especificado, se toma por defecto la dirección 192.168.0.254.

- SYSLOG_HOST: Si se requiere enviar los mensajes de sistema a una máquina distinta al servidor por defecto, entonces se tiene que especificar dicha máquina aquí. Si ningún parámetro es especificado se utilizará la entrada de SERVER.

- NFS_SERVER: Especifica la dirección IP del servidor desde donde el sistema de archivos /home es montado. Por defecto se utiliza la entrada SERVER.

- USE_NFS_SWAP: Esta opción habilita el uso de memoria SWAP sobre NFS. Por defecto es N.

- SWAPFILE_SIZE: Permite controlar el tamaño del archivo de intercambio. El tamaño por defecto es 64MB.

- SWAP_SERVER: El archivo de intercambio puede existir en cualquier servidor de la red. Se puede especificar la dirección IP de dicho servidor. Por defecto se toma el valor de NFS_SERVER.

- NFS_SWAPDIR: Especifica el directorio que es exportado vía NFS. Por defecto es /var/opt/ltsp/swapfiles. Es necesario que dicho directorio esté registrado en el archivo /etc/exports.

- TELNET_HOST: Si la terminal es configurada para tener una sesión en modo texto, entonces el valor de este parámetro será usado como el host donde se haga la sesión de telnet. Si ningún valor es especificado se utilizará la entrada de SERVER.

- DNS_SERVER: Usada para crear el archivo resolv.conf.

- SEARCH_DOMAIN: Usada para crear el archivo resolv.conf.

MODULE_01 hasta MODULE_10

10 módulos de kernel pueden ser cargados mediante estas entradas. Por ejemplo:

```
MODULE_01 = uart401.o
MODULE_02 = sb.o io=0x220 irq=5 dma=1
MODULE_03 = opl3.o
```

Si el valor de este parámetro es una ruta absoluta, entonces el comando insmod será utilizado. De otro modo, se utilizará modprobe.

- RAMDISK_SIZE: Cuando la terminal se inicia, crea un disco RAM y lo monta en el directorio /tmp. Se puede controlar el tamaño de este sistema de archivos con este parámetro. Se debe especificar las unidades en kilobytes (1024 bytes). Por ejemplo para crear un disco RAM de 2 MB se debe especificar RAMDISK_SIZE=2048.

- RCFILE_01 hasta RCFILE_10: Algunos scripts adicionales pueden ser ejecutados por rc.local. Se debe colocar el script en el directorio /etc/rc.d y especificar el nombre en alguna de estas entradas.

- SOUND: Si el paquete LTSP Sound está instalado, se necesita poner el valor 'Y' en esta entrada y ejecutar el script rc.sound para configurar la tarjeta de sonido y el demonio. Por defecto es N.

- RUNLEVEL: El runlevel determina el modo en el cual la terminal estará corriendo. Los siguientes runlevels son los soportados:

  3: Iniciará un shell. Útil para depuración de la terminal.
  4: Correrá una o más sesiones de telnet en el TELNET_HOST.
  5: Modo GUI. Arrancará X Window, y enviará una petición XDMCP al servidor, quien devolverá una pantalla de login a la terminal. Se necesita tener un Display Manager corriendo en el servidor, como XDM, GDM o KDM.

- TELNET_SESSIONS: Indica cuántas sesiones de telnet se pueden correr. Cada sesión estará en una diferente pantalla virtual, a las que se puede acceder con las teclas ALT-F1 a ALT-F9. El valor por defecto es 2.

B) Opciones de X Window

- XDM_SERVER: Si se requiere que XDM corra en una máquina que no sea el servidor por defecto, entonces se debe especificar aquí. Si este parámetro no es especificado, entonces se usará la entrada en SERVER.

- XSERVER: Esta entrada define qué servidor X correrá la terminal. Para placas PCI y AGP, este parámetro no debería ser necesario. El script rc.local debería autodetectar la tarjeta. También se puede activar auto a fin de indicar que se tratará de autodetectar el video. Para placas ISA, o para especificar un servidor X en particular, se puede poner el nombre del driver o servidor X. Si el valor comienza con 'XF86_', entonces XFree 3.3.6 será utilizado, sino, XFree 4.1.x lo será. El valor por defecto es auto.

- X_MODE_0 hasta X_MODE_2 : Hasta 3 resoluciones pueden ser configuradas para la terminal. Esta entrada puede tomar dos tipos diferentes de valores. Puede ser tanto una resolución como un modeline completo:

X_MODE_0 = 800x600 o bien

X_MODE_0 = 800x600 60.75 800 864 928 1088 600 616 621 657 -HSync –Vsync

Si ninguna de las entradas X_MODE_x es especificada, entonces se utilizarán los modelines por defecto, y las resoluciones serán de 1024x768, 800x600 y 640x480. Si una o más entradas X_MODE_x es especificada, entonces no se tendrán en consideración los modelines por defecto.

- X_MOUSE_PROTOCOL: Cualquier valor que funcione con el protocolo de puntero XFree86 puede ser especificado. Los valores típicos incluyen 'Microsoft' y 'PS/2'. El valor por defecto es 'PS/2'.

- X_MOUSE_DEVICE: Este es el dispositivo al cual el mouse está conectado. Si es un mouse serial, debería ser un puerto serie, como /dev/ttyS0 o /dev/ttyS1. Si es un mouse PS/2, este valor debería ser /dev/psaux. El valor por defecto es /dev/psaux.

- X_MOUSE_RESOLUTION: Este es el valor 'Resolution' que se encuentra en cualquier archivo XF86Config. Un valor típico para un mouse serial es 50 y para uno PS/2 es 400. El valor por defecto es 400.

- X_BUTTONS: Indica al sistema cuántos botones tiene el mouse, usualmente son 2 o 3 y el valor por defecto es 3.

- X_MOUSE_EMULATE3BTN: Esto le indica al servidor X que debe emular el tercer botón del mouse cuando se presionen simultáneamente los botones izquierdo y derecho. El valor por defecto es N.

- X_MOUSE_BAUD: Para ratones seriales, define la tasa de baudios. El valor por defecto es 1200.

- X_COLOR_DEPTH: Este es el número de bits a utilizar para la profundidad del color. Los valores posibles son 8, 15, 16, 24 y 32. 8 bits darán 256 colores, 16 - 65536, 24 - 16 millones y 32 - 4,2 billones de colores. No todos los servidores X soportan estos valores. El valor por defecto es 16.

- USE_XFS: Se tiene la opción de correr un Servidor de Fuentes X (XFS, X Font Server) o bien leer las fuentes vía NFS. El servidor de fuentes debería proveer un camino simple para mantener todas las fuentes en un solo lugar, pero hay algunos problemas cuando el número de terminales supera las 40. Los 2 valores en esta opción son Y o N, el valor por defecto es N. Si se desea usar un Servidor de Fuentes, entonces se debe utilizar la entrada XFS_SERVER para especificar su dirección IP.

- XFS_SERVER: Si se está usando un Servidor de Fuentes, se debe especificar su dirección IP. Si no está especificado, entonces se usará por defecto la entrada especificada en SERVER.

- X_HORZSYNC: Define la sincronización horizontal del monitor y el valor por defecto es "31-62".

- X_VERTREFRESH: Define con que frecuencia se refresca verticalmente el monitor. Por defecto es "55-90".

- XF86CONFIG_FILE: Si se desea tener un archivo propio de configuración XF86Config, se puede especificar en esta opción. Sólo se tiene que colocar en el directorio /opt/ltsp/i386/etc. No importa su nombre, siempre y cuando dicho nombre sea indicado en éste parámetro. Por ejemplo:

- XF86CONFIG_FILE = XF86Config.ws004: Los scripts de configuración del LTSP crean este archivo el cual tiene la siguiente estructura:

```
    # Keyboards
    XkbSymboles     = "us(pc101)"
    XkbModel        = "pc101"
    XkbLayout       = "us"

    USE_XFS         = N
    LOCAL_APPS      = N
    RUNLEVEL        = 5

    # uncomment the following line to enable floppy support
    #RCFILE_01      = floppyd

    # uncomment the following line to enable USB support
    #RCFILE_02      = usb

    # enable sound by default
    SOUND           = Y

    # default sound volume
    VOLUME          = 75

    ### For ISA sound cards, you have to specify the module to use:
    # SMODULE_01    = sb io=0x220 irq=5 dma=1
#-----------------------------------------------------------------------
# Below are sample mode lines for a variety of vertical refresh rates and
# resolutions.  They are used to define the default client screen resolution.
# Some lines may not work with a given monitor and video chipset.
# To avoid damaging a monitor and video card, only specify mode lines that
# your client's hardware can support.
# Uncomment only *one* of the following X_MODE_0 lines at a time, or add one
# of your own.
```

```
# 60 Hz Resolutions
#  X_MODE_0 = 640x480  25.175  640  656  752  800   480 490 492 525  -hsync -vsync
#  X_MODE_0 = 800x600  40      800  840  968 1056   600 601 605 628  +hsync +vsync
#  X_MODE_0 = 1024x768 65     1024 1048 1184 1344   768 771 777 806  -hsync -vsync

# 70 Hz Resolutions (Use instead of 72 Hz for 1024x768)
#  X_MODE_0 = 1024x768 75     1024 1048 1184 1328   768 771 777 806  -hsync -vsync

# 72 Hz Resolutions
#  X_MODE_0 = 640x480  31.5    640  664  704  832   480 489 492 520  -hsync -vsync
#  X_MODE_0 = 800x600  50      800  856  976 1040   600 637 643 666  +hsync +vsync
#  X_MODE_0 = 1024x768 75     1024 1048 1192 1296   768 771 777 806  -hsync -vsync

# 75 Hz Resolutions
#  X_MODE_0 = 800x600  49.5    800  816  896 1056   600 601 604 625  +hsync +vsync

# 85 Hz Resolutions
#  X_MODE_0 = 800x600  60.75   800  864  928 1088   600 616 621 657  -hsync -vsync

#----------------------------------------------------------------------
#
# Example of specifying X settings for a workstation
#
[ws001]
    XSERVER         = auto
    LOCAL_APPS      = N
    USE_NFS_SWAP    = N
    SWAPFILE_SIZE   = 48m
    RUNLEVEL        = 5

[ws002]
    XSERVER         = XF86_SVGA
    LOCAL_APPS      = N
    USE_NFS_SWAP    = N
    SWAPFILE_SIZE   = 64m
    RUNLEVEL        = 3

#
# ws004 is my virtual workstation running in a VMware session
#
[ws004]
    DNS_SERVER      = 192.168.0.254
    XSERVER         = auto
    X4_BUSID        = "PCI:0:15:0"
    X_MODE_0        = 800x600
    LOCAL_APPS      = N
    USE_NFS_SWAP    = N
    SWAPFILE_SIZE   = 64m
    RUNLEVEL        = 5

#----------------------------------------------------------------------
#
# Example of a workstation configured to load some modules
```

```
#
#[ws001]
#   MODULE_01      = agpgart.o          # This is for i810 video
#   MODULE_02      = uart401.o
#   MODULE_03      = sb.o io=0x220 irq=5 dma=1
#   MODULE_04      = opl3.o

#-----------------------------------------------------------------------
#
# Example of ws001 configured for local apps
#
#[ws001]
#   LOCAL_APPS     = Y
#   LOCAL_WM       = Y
#   NIS_DOMAIN     = ltsp
#   NIS_SERVER     = 192.168.0.254

#-----------------------------------------------------------------------
#
# Example of a serial printer attached to /dev/ttyS1 on workstation ws001
#
#[ws001]
#   PRINTER_0_DEVICE  = /dev/ttyS1
#   PRINTER_0_TYPE    = S          # P-Parallel, S-Serial
#   PRINTER_0_PORT    = 9100       # tcp/ip port: defaults to 9100
#   PRINTER_0_SPEED   = 9600       # baud rate: defaults to 9600
#   PRINTER_0_FLOWCTRL = S         # Flow control: S-Software (XON/XOFF),
#                        # H-Hardware (CTS/RTS)
#   PRINTER_0_PARITY  = N          # Parity: N-None, E-Even, O-Odd
#                        # (defaults to 'N')
#   PRINTER_0_DATABITS = 8         # Databits: 5,6,7,8 (defaults to 8)
```

Tabla 3-9 Ejemplo de configuración del archivo lts.conf

3.2.5.6.1 Terminal 1

Para este equipo, la configuración del archivo lts.conf debe ser:

```
[ws001]
        XSERVER             = Xvesa
        LOCAL_APPS          = N
        USE_NFS_SWAP        = N
        SWAPFILE_SIZE       = 48m
        RUNLEVEL            = 5
        X_HORZSYNC          = "30.9-49.0"
        X_VERTREFRESH       = "50.0-100.0"
        X_MODE_0            = 800x600
```

3.2.5.6.2 Terminal 2

Para este equipo, la configuración del archivo lts.conf debe ser:

```
[ws002]
    XSERVER          = Xvesa
    LOCAL_APPS       = N
    USE_NFS_SWAP     = N
    SWAPFILE_SIZE    = 48m
    RUNLEVEL         = 5
    X_HORZSYNC       = "30.9-49.0"
    X_VERTREFRESH    = "50.0-100.0"
    X_MODE_0         = 800x600
```

3.2.5.6.3 Terminal 3

Para este equipo, la configuración del archivo lts.conf debe ser:

```
[ws003]
    XSERVER          = Xvesa
    LOCAL_APPS       = N
    USE_NFS_SWAP     = N
    SWAPFILE_SIZE    = 48m
    RUNLEVEL         = 5
    X_MODE_0         = 800x600
```

3.3 Terminales

Una vez que se haya terminado de instalar y configurar todos los paquetes necesarios en el servidor, el siguiente paso es crear una imagen de sistema operativo para que las terminales puedan arrancar. Donde las posibilidades de arranque podrían ser las siguientes:

- Arranque desde la tarjeta de red: Las tarjetas necesitan un chip EPROM para grabar en él la imagen de arranque. Arranque desde la propia BIOS si se incorpora PXE (Preboot-eXecution Environment).

- Arranque desde disquete: La imagen de arranque se graba en un disquete 3 ½ " de alta densidad. Esta es la solución más económica y sencilla.

- Arranque desde el disco duro: Configurando el archivo Config.sys y Autoexec.bat en WINDOWS.

3.3.1 Creación del disquete de arranque

Es importante mencionar que no basta con crear una imagen de arranque común, la diferencia radica en que ésta deberá estar etiquetada especialmente para que el kernel permita arrancar la terminal vía red.

La forma más común es bajar la imagen de arranque y copiarla en un disquete[13]. Se deberá elegir la opción Floppy Bootable Rom Image, lo cual generará una imagen que se puede escribir en un disquete con los comandos dd o cat, desde Linux; o rawrite en MS-DOS.

Este disquete primero intentará detectar la tarjeta de red y luego iniciará el proceso de arranque como si el código estuviera en una eprom desde la misma tarjeta de red. Por ejemplo, para generar el disquete de arranque para la tarjeta MX98715 de la terminal 3, se tiene que hacer lo siguiente:

- Ir al sitio y hacer click en la liga de la versión 5.0.7.
- Seleccionar el tipo de tarjeta de red, en este caso es "mx98715".
- Elegir el formato de salida Floppy Bootable ROM Image (.lzdsk).
- Hacer click en "Get ROM" para generar la imagen de arranque. Se creará el siguiente archivo:

```
eb-5[1].0.7-mx98715.lzdsk
```

Para crear el disco de arranque no basta con copiar la imagen al disquete, es necesario que éste sea booteable para esto se utiliza cualquiera de los siguientes comandos:

En Linux se utiliza:

```
cat eb-5[1].0.7-mx98715.lzdsk > /dev/fd0
```

Esto copiará la imagen en el primer sector del disquete.

[13] Página del sitio para descargar la imagen de arranque http://www.rom-o-matic.net

O desde MS-DOS, usando el comando rawrite, el cual viene incluido en el primer disco de instalación de Fedora Core, en la carpeta DOSUTILS.

Se escribe el comando:

```
D:\DOSUTILS\rawrite
```

Se pedirá la ruta de la imagen de arranque:

```
C:\eb-5[1].0.7-mx98715.lzdsk
```

Y después la ruta donde se escribirá la imagen de arranque, en este caso es el disquete:

```
A:\
```

Este comando no acepta archivos con extensiones mayores a tres caracteres, por lo tanto se tendrá que cambiar la extensión del archivo, en este caso será .img.

```
C:\ren eb-5[1].0.7-mx98715.lzdsk   mx98715.img
```

Y se tendrá que repetir el proceso con el comando rawrite hasta que se haya logrado grabar correctamente la imagen en el disquete.

Después de crear el disco de arranque, será necesario probar que éste reconoce la tarjeta de red de la terminal, lo cual se puede comprobar si al bootear la terminal con el disquete se muestra la MAC address de la tarjeta y aparece un mensaje que indica que se está buscando un servidor DHCP.

3.3.2 Arranque por medio del disco duro

Otra alternativa para el arranque de las terminales es la creación de un menú en Windows en el disco duro local, lo cual permite iniciar Linux sin la necesidad de utilizar un disquete de arranque y a su vez iniciar Windows normalmente.

Es importante mencionar que esté método solo funciona en Windows 95 y 98, ya que en versiones posteriores el soporte para el menú del archivo config.sys fue

deshabilitado. Las terminales 1 y 2 tienen instalado Windows 95 por lo que utilizarán este método, en cambio la terminal 3 tiene instalado Windows Millenium, por lo que se utilizará un disquete de arranque.

Para crear el menú se requiere hacer lo siguiente:

- Ir al sitio[14] y hacer click en la liga de la versión 5.0.7.
- Seleccionar el tipo de tarjeta de red, en este caso es "3c509" para las terminales 1 y 2.
- Elegir el formato de salida DOS Executable ROM Image (.com).
- Hacer click en "Configure" para definir que en el proceso de arranque se muestre una barra giratoria en lugar de una sucesión de puntos.
- Hacer click en "Get ROM" para generar la imagen de arranque. Se creará el siguiente archivo:

```
eb-5[1].0.7-3c509.com
```

Se debe renombrar el archivo generado con un nombre descriptivo, no mayor a 8 caracteres. Esto se puede hacer directamente en el explorador de Windows o en la línea de comandos de MS-DOS.

```
C:\ren  eb-5[1].0.7-3c509.com   3c509.com
```

Posteriormente se debe crear un directorio directamente en C:\ llamado ltsp, donde se colocará el archivo 3c509.com.

```
C:\md ltsp
C:\cp 3c509.com  ltsp\
```

Se tiene que editar el archivo config.sys y agregar lo siguiente al inicio:

[14] Página del sitio para crear al menú de inicio http://www.rom-o-matic.net

```
[Menu]
MenuItem=windows,Microsoft Windows 95/98
MenuItem=linux,Linux Terminal Server Project
MenuDefault=windows,10

[linux]
DOS=NOAUTO

[windows]
```

Esto creará el menú con las opciones de inicio Windows y Linux.

También se tiene que editar el archivo autoexec.bat agregando lo siguiente al inicio:

```
@ECHO OFF
GOTO %CONFIG%

:linux
C:\ltsp\3c509.com

:windows
```

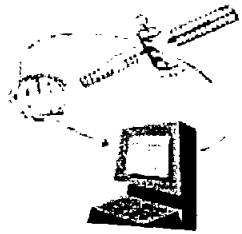
Esto permitirá realmente ejecutar la imagen de arranque una vez que se haya seleccionado la opción Linux en el menú de inicio.

Hasta éste punto preparamos el equipo para instalar adecuadamente el sistema operativo así como los diferentes componentes que intervienen en la implementación; fueron instalados los archivos de sistema y aprendimos a crear el disco de arranque. Ahora estamos listos para iniciar todo el sistema, analizaremos el inicio de una terminal para lograr detectar problemas y con lo que hemos visto en los capítulos anteriores podremos instalar sin problemas el software adicional así como la configuración a internet.

PUESTA EN MARCHA

OBJETIVO ESPECÍFICO

Aplicar los conocimientos teóricos y prácticos para garantizar el buen funcionamiento del servidor de terminales gráficas en el momento de iniciar todo el sistema. Entender el proceso de cómo inicia una terminal para poder detectar y darle la solución más óptima a los posibles problemas.



4

INTRODUCCIÓN

Hasta este punto se han instalado los paquetes del LTSP y del software necesario, también se han realizado las configuraciones correspondientes a los archivos de sistema; sin embargo falta comprobar que efectivamente cada una de las terminales inicien una sesión gráfica debido a que estas configuraciones no garantizan que dichas terminales funcionen correctamente, para ello se cuenta con un apartado de detección y solución de problemas más comunes.

Una vez que han sido solucionados los problemas y se han iniciado las terminales gráficamente estarán listas para ser conectadas en red de forma local y a su vez tendrán acceso a la red global.

Tomando en cuenta que en el laboratorio se utiliza software comercial de electrónica, se hizo una recopilación de software de este tipo, pero que puede correr bajo Linux y con la ventaja de que es libre.

4.1 Teoría de operación

El arranque de una terminal gráfica aplicando el LTSP involucra muchos pasos, es importante conocer y comprender de forma general cada uno de ellos; ya que si en algún momento surgen problemas, estos podrían solucionarse más rápidamente, debido a que se puede deducir el origen de ellos.

El proceso de arranque de una terminal aplicando el LTSP es el siguiente:

- Cuando se enciende la terminal, ésta busca entre los dispositivos de almacenamiento disponibles un sistema operativo que iniciar; cuando detecta el código de arranque, éste se empezará a ejecutar.

- El código de inicio tratará de detectar una tarjeta de red y una vez que lo hace, la inicializa.

- Se hará una petición de broadcast buscando un servidor dhcp en la red local, esta petición incluirá la dirección MAC de la tarjeta de red.

- El proceso inetd del servidor verá la petición de broadcast e invocará al demonio dhcpd para responderla.

- El demonio dhcpd leerá el archivo de configuración, dhcpd.conf, y tratará de localizar la entrada que corresponda con la dirección MAC de la terminal. Al ser encontrada, la pondrá en un paquete de respuesta y se lo enviará a la terminal que pidió la información. Ciertos parámetros serán incluidos en este paquete, siendo los más importantes:

    - Dirección IP asignada a la terminal.
    - Máscara de subred de la red de área local.
    - Directorio home del archivo de inicio.
    - Nombre del kernel a bajar.

- El código de inicio recibirá el paquete de respuesta dhcp y configurará la interfaz TCP/IP en la tarjeta de red con los parámetros proporcionados.

- El código de inicio enviará una petición TFTP al servidor para empezar a bajar el kernel.

- Una vez que el kernel fue completamente bajado por la terminal, éste tomará el control.

- El kernel empezará a ejecutarse, inicializando todo el sistema y sus periféricos.

- El código de inicio le puede pasar información al kernel y lo hace construyendo una línea de comandos, similar a la manera en que LILO o GRUB le pasa parámetros.

- Si el kernel necesita hacer una petición dhcp, el servidor responderá con otro paquete, conteniendo la información que el kernel necesita para continuar. Los parámetros importantes en esta respuesta son:

    - Dirección IP asignada a la terminal.
    - Máscara de subred de la red de área local.
    - El directorio raíz a ser montado vía NFS.
    - La puerta de enlace (gateway).
    - El servidor de nombres (DNS).
    - El nombre de host de la terminal.

- El directorio raíz de la terminal será montado vía NFS. Este sistema de archivos será de sólo lectura; porque se pueden tener varias terminales montando el mismo sistema de archivos, y con esto se evita que en alguna de ellas se modifiquen los contenidos del mismo.

- En este punto, el control será pasado del kernel al proceso init.

- Init leerá el archivo inittab y empezará a configurar el ambiente.

- Uno de los primeros elementos de inittab es el archivo rc.local, que correrá mientras la terminal está en el estado 'sysinit'.

- El script en rc.local creará un disco RAM de 1 MB que contendrá todas las cosas que necesitan ser escritas o modificadas de alguna manera.

- Este disco RAM será montado en el directorio /tmp. Cualquier archivo que necesite ser escrito estará en el directorio /tmp, con un enlace simbólico apuntando hacia él.

- El directorio /proc es montado.

- La interfaz de loopback es configurada.

- Varios directorios son creados dentro de /tmp para que contengan algunos de los archivos transitorios que son necesarios mientras el sistema está corriendo. Algunos de ellos son:

    - /tmp/compiled
    - /tmp/var
    - /tmp/var/run

- /tmp/var/log
- /tmp/var/lock
- /tmp/var/lock/subsys

- El archivo XF86Config es generado y está basado en las entradas del archivo lts.conf. Aquí es donde la información acerca del tipo de mouse, y otros parámetros para el servidor X son combinados para crear el archivo de configuración.

- El script start_ws es creado, éste es responsable de iniciar el servidor X y cuando lo inicia, es dirigido para enviar peticiones XDMCP al servidor XDM. Esto funciona basado en la información encontrada en el archivo lts.conf.

- El archivo syslog.conf es creado, éste archivo contiene información que le indica al demonio syslogd a qué host de la red le deberá enviar la información de log. Este host de syslog está especificado en el archivo lts.conf. Hay un enlace simbólico llamado /etc/syslog.conf que apunta a /tmp/syslog.conf.

- El demonio syslogd se inicia, usando el archivo de configuración recientemente creado.

- El control es pasado otra vez a init y buscará en la entrada initdefault para saber a cuál runlevel entrar, por lo general el runlevel por defecto es 2.

- El runlevel 2 hará que se ejecute el script set_runlevel, que leerá el archivo lts.conf para saber en qué modo correr la terminal.

- Los niveles de ejecución estándar de LTSP son 3, 4 y 5.

  - El nivel 3 iniciará en un shell en la terminal.
  - El nivel 4 ejecutará una o más sesiones de telnet en el servidor. Esto es ideal para remplazar viejas terminales de tipo serial.
  - El nivel 5 entrará en el modo GUI, levantando el cliente X y haciendo una petición XDMCP al servidor, quien mandará una ventana de diálogo de inicio para permitir al usuario autenticarse. Hace falta un Display Manager corriendo en el servidor, como XDM, GDM o KDM.

4.2 Iniciando las terminales

La estrategia que se adoptó fue probar de manera independiente cada una de las terminales, para lo cual es necesario conectarlas directamente al servidor por medio de un cable cruzado; la ventaja de hacerlo de esta forma es que se tiene un mejor control en cuanto a la posible detección y solución de problemas.

Una vez que se haya comprobado que todas las terminales inician correctamente el sistema operativo y una respectiva sesión gráfica, se podrá implementar realmente la red, es decir; las terminales se conectarán al switch y éste a su vez al servidor, para lo cual será necesario utilizar cable directo en lugar de cable cruzado.

Asumiendo que las configuraciones del servidor fueron las adecuadas, al iniciar una sesión será desplegada la siguiente pantalla:

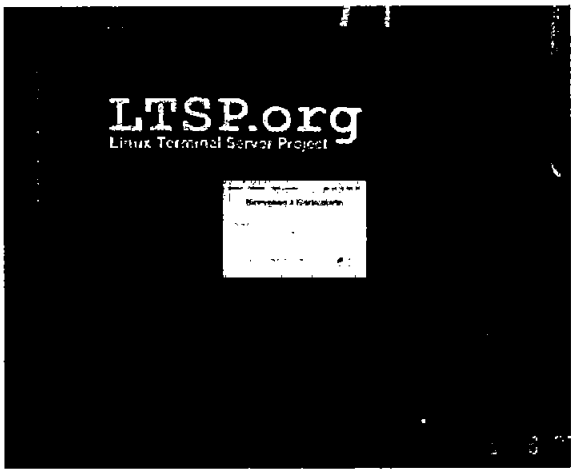


Figura 4-1 Pantalla de bienvenida del LTSP

Debido a su importancia, se describirá en la siguiente sección los problemas que se detectaron así como otros posibles errores en la implementación del LTSP, además la forma en que pueden ser solucionados.

4.3 Detección y solución de problemas

Los scripts de inicialización del LTSP hacen un gran trabajo al realizar ciertas configuraciones en los archivos necesarios, pero debido a la gran diversidad de hardware disponible para las terminales, es poco probable que éstas inicien completamente y de manera correcta la primera vez; es por eso que se debe comenzar con un proceso de detección y solución de problemas.

Conforme se va teniendo más experiencia en el proyecto LTSP, resulta más sencillo detectar los posibles errores y determinar la solución apropiada en cada caso; para ello es necesario observar hasta que punto ha llegado la terminal en su proceso de inicio.

A continuación se presentan los problemas más comunes, la manera de cómo detectarlos y sus posibles soluciones.

4.3.1 Conexiones

Es probable que la terminal no esté conectada apropiadamente a la red local, sobre todo cuando apenas se está implementando dicha red y ni siquiera se ha probado la conectividad entre las máquinas.

Para verificar esto, hay que observar que las luces de la tarjeta de red estén encendidas así como para todas las conexiones de red. Esto garantiza que al menos existe la conectividad entre la terminal y el dispositivo de la red local correspondiente.

En algunas ocasiones por cuestiones de prueba, es conveniente trabajar con la terminal conectada directamente al servidor; para ello es necesario que entre la terminal y el servidor (sin hacer uso de un hub o switch) se esté utilizando un cable cruzado.

Si se está utilizando un hub o switch, entonces hay que asegurarse que se está usando un cable directo, tanto entre las terminales y el hub, como entre el hub y el servidor.

En el capítulo 2 se muestra la configuración de los pines para construir un cable directo y un cable cruzado.

4.3.2 Imagen de arranque

La imagen de arranque debe reconocer la tarjeta de red de la terminal para poderla inicializar y configurar correctamente; si esta imagen no corresponde a la tarjeta de red, entonces no servirá para inicializar dicha terminal.

Cuando se inicia la terminal por medio de la imagen de arranque, se debe observar una salida como la siguiente:

```
loaded ROM segment 0x0800 length 0x4000 reloc 0x9400
Etherboot 5.0.1 (GPL) Tagged ELF for [LANCE/PCI]
Found AMD Lance/PCI at 0x1000, ROM address 0x0000
Probing...[LANCE/PCI] PCnet/PCI-II 79C970A base 0x1000, addr 00:50:56:81:00:01
Searching for server (DHCP)...
<sleep>
```

Tabla 4-1 Mensaje de salida cuando la imagen de arranque es apropiada

Para identificar si la imagen de arranque es la apropiada, ésta debe mostrar la dirección MAC de la tarjeta de red, sin embargo esto no garantiza que funcione completamente.

Si, por el contrario, se obtiene una salida como la siguiente, entonces es probable que la imagen de arranque generada no sea la apropiada para la tarjeta de red ya que no detecta ni muestra la dirección MAC.

```
ROM segment 0x0800 length 0x8000 reloc 0x9400
Etherboot 5.0.2 (GPL) Tagged ELF for [Tulip]
Probing...[Tulip]No adapter found
<sleep>
<abort>
```

Tabla 4-2 Mensaje de salida al iniciar la terminal con la imagen de arranque

Si no se muestran estos mensajes indicando que Etherboot a iniciado, entonces puede ser que el disquete esté dañado o que se haya grabado la imagen de arranque de manera incorrecta.

En el capítulo 3 se indica la manera de generar la imagen de arranque, así como la forma de grabarla en el disquete.

4.3.3 Problemas con el servicio DHCP

Una vez que la tarjeta de red es inicializada, el código de la imagen de arranque enviará una petición de broadcast a través de la red local, buscando algún servidor DHCP. Si la terminal obtiene una respuesta válida de parte del servidor DHCP, deberá configurar la tarjeta de red.

Si el servidor DHCP está funcionando correctamente, éste responderá a la terminal enviándole un paquete con la información que está definida en su archivo de configuración (dhcpd.conf) y que corresponde a la terminal que envió la petición.

```
ROM segment 0x0800 length 0x4000 reloc 0x9400
Etherboot 5.0.1 (GPL) Tagged ELF for [LANCE/PCI]
Found AMD Lance/PCI at 0x1000, ROM address 0x0000
Probing...[LANCE/PCI] PCnet/PCI-II 79C970A base 0x1000, addr 00:50:56:81:00:01
Searching for server (DHCP)...
<sleep>
Me: 192.168.0.1, Server: 192.168.0.254, Gateway 192.168.0.254
```

Tabla 4-3 Mensaje de que el servidor DHCP funciona correctamente

Si se muestra la línea que comienza con 'Me:', seguida por una dirección IP, entonces es un hecho que DHCP está funcionando correctamente. En ésta línea se mostrará la dirección IP asignada a la terminal, y las direcciones IP del servidor y del gateway.

Si por el contrario se muestra el siguiente mensaje en la terminal y no se obtiene respuesta por parte del servidor, es probable que haya un problema con el servicio DHCP.

```
Searching for server (DHCP)...
<sleep> <sleep>
```

Tabla 4-4 Mensaje de que quizá no funciona adecuadamente el servicio DHCP

Hay que notar que el mensaje <sleep> se puede mostrar por algunos segundos, debido al tráfico en la red, o porque por alguna razón el servidor estaba muy ocupado y no había podido atender la petición de la terminal; sin embargo no es normal que el servidor tarde mucho tiempo en dar una respuesta.

En el caso de que el servidor DHCP no responda, se tiene que averiguar las posibles causas; entre las cuales se encuentran las siguientes.

4.3.3.1 El demonio dhcpd

Para determinar si el demonio dhcpd está corriendo en el servidor se pueden ejecutar algunos comandos, e interpretar su salida; dhcpd normalmente corre en modo background escuchando en el puerto udp 67.

Se debe ejecutar el comando netstat para ver si hay algo escuchando en dicho puerto:

```
netstat -an | grep ":67 "
```

Se debe mostrar una salida similar a la siguiente:

```
udp    0      0      0.0.0.0:67      0.0.0.0:*
```

Tabla 4-5 Mensaje para saber si el demonio dhcpd esta corriendo en el servidor

La cuarta columna contiene la dirección IP y el puerto, separadas por dos puntos ":". Una dirección con solo ceros "0.0.0.0" indica que se está escuchando en todas las

interfases. Esto significa que se pueden tener varias tarjetas de red en el servidor; una interfaz eth0, otra eth1, etc., y que dhcpd esté escuchando en todas ellas.

Aunque se haya mostrado que existe un servicio escuchando en el puerto udp 67, esto no significa que efectivamente dhcpd sea el que lo esté haciendo; ya que BOOTP el cual es un servicio similar a DHCP, podría ser el que estuviera ocupando el puerto udp 67.

Actualmente en la mayoría de las distribuciones Linux solo se incluye el servicio DHCP, pero para asegurarse que el demonio esté ejecutándose realmente en el servidor, se tiene que teclear el siguiente comando:

```
ps aux | grep dhcpd
```

La salida es parecida a la siguiente:

```
root 23814 0.0 0.3 1676 820 ? S 15:13 0:00 /usr/sbin/dhcpd
root 23834 0.0 0.2 1552 600 pts/0 S 15:52 0:00 grep dhcp
```

Tabla 4-6 Mensaje para asegurarse que el demonio dhcpd esté ejecutándose

En realidad la primera línea es la que muestra que dhcpd está corriendo, la segunda es tan sólo el comando grep que se utilizó para hacer la búsqueda en la salida del comando ps.

Si no se muestra ninguna línea indicando que dhcpd está corriendo, entonces se necesita verificar que el servidor esté configurado para el runlevel 5 y que dhcpd esté listo para iniciar en dicho nivel. En sistemas basados en Fedora Core, se puede ejecutar el programa ntsysv para asegurar que el servicio DHCP esté seleccionado en el nivel de ejecución apropiado.

Existe una alternativa al comando ps, para saber si se está ejecutando dhcpd; se debe teclear lo siguiente:

```
service dhcpd status
```

Este comando mostrará el estado del servicio (si está detenido o en ejecución); si está ejecutándose indicará los ID de proceso que está utilizando.

```
Se está ejecutando dhcpd (pid 3693)...
```

Tabla 4-7 Mensaje que muestra el estado del servicio dhcpd

Si efectivamente el demonio dhcpd está detenido o con algún problema; se puede probar iniciando o reiniciando el servicio con estos comandos:

```
service dhcpd start
service dhcpd restart
```

En algunas ocasiones cuando se intenta iniciar dhcpd, aparece el siguiente mensaje:

```
st decls to leases file.
Wrote 0 new dynamic host decls to leases file.
Wrote 0 leases to leases file.
Not configured to listen on any interfaces!
```

Tabla 4-8 Mensaje de error de que no están activas las tarjetas de red

Este mensaje de error significa que no existen interfaces ethernet en el servidor o que no están activadas, esto se soluciona con el programa de configuración apropiado; en el caso de Fedora Core existe una herramienta llamada redhat-config-network, con ella se pueden configurar y activar las tarjetas de red del sistema.

Una vez que se haya(n) configurado la(s) tarjeta(s) de red, proporcionando su dirección IP, máscara de red, puertas de enlace, etc., y después de haberla(s) activado; se puede comprobar su estado con el comando ifconfig, el cual mostrará las interfaces ethernet activadas. En este momento, podrá iniciarse el demonio dhcpd sin ningún problema.

Si los comandos start o restart no arrojan errores, significa que dhcpd ahora sí, estará corriendo en el servidor.

4.3.3.2 Verificar la configuración de dhcpd.conf

Como se vio en el capítulo 3, el archivo dhcpd.conf contiene las configuraciones necesarias para que el servicio DHCP funcione correctamente; en ocasiones basta con adaptar ciertas cosas al archivo de ejemplo dhcpd.conf.example; particularmente las direcciones MAC de las terminales. Sin embargo hay situaciones en las que se debe modificar otros parámetros.

En primer lugar se debe verificar que la siguiente línea se encuentre antes del parámetro default-lease-time:

ddns-update-style none;

Esta línea es requerida a partir de la versión 3.0 de DHCP.

Hasta el momento en que la terminal necesita recibir la información para configurar la tarjeta de red y antes de comenzar a bajar el kernel desde el servidor, solo son requeridos ciertos parámetros del archivo dhcpd.conf.

Es necesario especificar las direcciones IP del servidor LTSP, del gateway y del servidor de dominio; en ambientes LTSP generalmente se utiliza un solo servidor que realice estas tres funciones, es decir una misma dirección IP será indicada en estos parámetros.

También se debe verificar que la entrada "hardware ethernet" concuerda exactamente con la dirección MAC de la terminal; ésta dirección se puede conocer cuando se inicia la terminal con la imagen de arranque apropiada.

hardware ethernet 00:E0:06:E8:00:84;

Finalmente se tienen que indicar las direcciones IP que serán asignadas a cada una de las terminales, estas se establecen en el parámetro "fixed-address".

fixed-address 192.168.0.1;

Con esta información bastará para que la terminal pueda configurar la tarjeta de red y comience a bajar la imagen del kernel desde el servidor LTSP.

4.3.3.3 Firewalls

Linux cuenta con herramientas llamadas firewalls, que disminuyen el riesgo de que algún intruso pueda comprometer la seguridad del sistema, sin embargo si estas no son configuradas adecuadamente para funcionar en conjunto con el ambiente LTSP, pueden ocasionar que el servidor no trabaje como se espera.

- Ipchains: Los kernels anteriores al 2.4 tenían la posibilidad de manipular la entrada y salida de los paquetes de datos usando ipchains, el cual utilizaba listas de reglas que se aplicaban a los paquetes en cada paso del proceso de filtrado.

Si en el servidor LTSP, se trabaja con un kernel cuya versión sea menor a la 2.4; se debe teclear el siguiente comando para determinar si ipchains está bloqueando la salida o entrada de paquetes:

```
ipchains -L –v
```

La salida debe ser similar a la siguiente:

```
Chain input (policy ACCEPT: 229714 packets, 115477216 bytes):
Chain forward (policy ACCEPT: 10 packets, 1794 bytes):
Chain output (policy ACCEPT: 188978 packets, 66087385 bytes):
```

Tabla 4-9 Mensaje que muestra si ipchains esta bloqueando los paquetes

En este caso se puede observar que ipchains está aceptando la entrada y salida de los paquetes de datos, por lo que no hay ningún conflicto entre dhcpd e ipchains.

- Iptables: Si la versión del kernel es mayor o igual a la 2.4, se tiene la posibilidad de manipular la entrada y salida de paquetes con ipchains o iptables, pero solo uno a la vez. Generalmente se utiliza iptables debido a que tiene mejoras en el funcionamiento y en el control disponible a la hora de filtrar paquetes.

Se debe ejecutar el siguiente comando para determinar el comportamiento de iptables:

```
iptables -L -v
```

La salida debe ser similar a la siguiente:

```
Chain INPUT (policy ACCEPT 18148 packets, 2623K bytes)
pkts bytes target prot opt in out source destination
Chain FORWARD (policy ACCEPT 0 packets, 0 bytes)
pkts bytes target prot opt in out source destination
Chain OUTPUT (policy ACCEPT 17721 packets, 2732K bytes)
pkts bytes target prot opt in out source destination
```

Tabla 4-10 Mensaje que muestra el comportamiento de iptables

Al igual que en ipchains, si se observa que se está aceptando la entrada y salida de paquetes, no habrá ningún problema entre iptables y dhcpd.

4.3.3.4 Problemas con la terminal

Si después de verificar las configuraciones en el servidor no se logra detectar en él algún problema, es posible que el conflicto se encuentre en la terminal. Para comprobar si la terminal está enviando la petición al servidor se puede observar el archivo /var/log/messages mientras la terminal se inicia.

Se debe teclear el siguiente comando:

```
tail -f /var/log/messages
```

Este comando mostrará las últimas líneas del archivo messages conforme se vayan agregando.

```
server dhcpd: DHCPDISCOVER from 00:50:56:81:00:01 via eth0
server dhcpd: no free leases on subnet WORKSTATIONS
server dhcpd: DHCPDISCOVER from 00:50:56:81:00:01 via eth0
server dhcpd: no free leases on subnet WORKSTATIONS
```

Tabla 4-11 Mensaje para saber si la terminal esta enviando la petición al servidor

Si se muestran algunas líneas en donde aparezca el mensaje "no free leases", entonces dhcpd está corriendo, pero en cambio hay un problema con la terminal debido a que no está haciendo la petición al servidor DHCP.

Esto puede deberse a que la imagen de arranque está defectuosa o la versión no es la correcta; aunque efectivamente dicha imagen corresponda a la tarjeta de red de la terminal, en algunas ocasiones la configuración con la que fue creada provoca que esta no funcione adecuadamente a la hora de arrancar la terminal.

Para solucionar esto, se debe probar iniciando la terminal con distintas versiones de la imagen de arranque; estas versiones se pueden conseguir también en la misma página[15].

En un caso extremo, si ninguna de las imágenes de arranque funciona, se puede probar reemplazando la tarjeta de red de la terminal; ya que puede tener algún problema físico que impida la conectividad con el servidor.

4.3.4 Problemas con el servicio TFTP

La imagen de arranque que permite iniciar las terminales contiene instrucciones o código Etherboot, el cual después de lograr comunicarse con el servidor utiliza el

[15] Página del sitio para descargar la imagen de arranque http://www.rom-o-matic.net.

protocolo TFTP para obtener un kernel Linux a través de la red. Este protocolo es muy simple, pero en ocasiones se generan ciertos problemas cuando se trata de hacerlo funcionar.

En el momento en que la terminal intenta bajar el kernel del servidor, se observa una línea similar a la siguiente:

```
Loading 192.168.0.254:/lts/vmlinuz.tulip .....
```

Tabla 4-12 Mensaje que muestra cuando la terminal intenta bajar el kernel

Posteriormente comienzan a desplegarse una serie de puntos que representan el avance de la descarga; mientras estos puntos continúen apareciendo en la pantalla significa que la imagen del kernel está siendo descargada exitosamente y por lo tanto TFTP está funcionando como se espera.

En el caso de que estos puntos no aparezcan de manera continua entonces quiere decir que hay un problema con el servicio TFTP, las posibles causas pueden ser las siguientes.

4.3.4.1 El demonio tftpd

En Fedora Core, tftp es iniciado por el demonio xinetd; existe un script de inicio llamado /etc/xinetd.d/tftp que contiene la información necesaria para que corra tftpd.

Es posible que el demonio tftpd no esté corriendo, por lo que se puede probar reiniciando el demonio xinetd, el cual reiniciará automáticamente un conjunto de servicios incluyendo tftp. Se debe teclear el siguiente comando:

```
service xinetd restart
```

4.3.4.2 La imagen del kernel no está en el directorio adecuado

La imagen del kernel que será descargado por la terminal, necesita estar en un lugar accesible para el demonio tftpd. Si la opción "-s" está especificada cuando tftpd inicia, entonces cualquier petición de la terminal debe ser relativa al directorio /tftpboot.

Por ejemplo, si el parámetro filename en el archivo de configuración dhcpd.conf es /lts/vmlinuz-2.4.19-ltsp-1, entonces la imagen del kernel deberá estar en la siguiente ruta: /tftpboot/lts/vmlinuz-2.4.19-ltsp-1.

Si no se especifica la opción "-s", entonces en el archivo dhcpd.conf tendrá que especificarse la ruta absoluta de la imagen del kernel, es decir:

```
filename        "/tftpboot/lts/vmlinuz-2.4.19-ltsp-1";
```

Tabla 4-13 Mensaje donde se especifica la ruta absoluta de la imagen del kernel

4.3.5 Problemas con el servicio NFS

NFS permite montar un sistema de archivos remoto a cada una de las terminales, éste sistema de archivos será realmente un directorio compartido que será exportado por el servidor; por lo que éste se debe configurar adecuadamente para que permita el acceso de las terminales.

Existen varios factores que pueden impedir que un sistema de archivos raíz pueda ser montado en las terminales:

4.3.5.1 El proceso Init no puede ser iniciado

Cuando la terminal intenta montar el sistema de archivos raíz, puede mostrar el siguiente error:

```
Kernel panic: No init found. Try passing init= option to kernel.
```

Tabla 4-14 Mensaje de error al intentar montar el sistema de archivos raíz

Este mensaje indica que el contenido del directorio exportado por el servidor no es el adecuado, o no contiene la información que necesita el kernel para continuar con el proceso de configuración del sistema.

Los scripts de configuración del LTSP pudieron haber tenido errores en la creación del directorio a exportar, por lo que es probable que el directorio /opt/ltsp/i386 esté vacío.

Se puede listar el contenido de este directorio para verificar que tenga la estructura correcta y la cual es la que espera encontrar el kernel de la terminal.

```
ls -l  /opt/ltsp/i386
```

Los siguientes directorios deberán estar presentes:

bin dev etc lib mnt oldroot opt proc root sbin tmp usr var

4.3.5.2 El servidor retorna el error -13

Si se muestra en la terminal el siguiente error:

```
Root-NFS: Server returned error -13 while mounting /opt/ltsp/i386
```

Tabla 4-15 Mensaje de error que muestra que el directorio i386 no esta en exports

Esto indicará que el directorio /opt/ltsp/i386 no está especificado en el archivo de configuración /etc/exports.

Es posible observar el contenido del archivo /var/log/messages para determinar alguna posible causa.

```
Jul 20 00:28:39 jamlap rpc.mountd: refused mount request from ws001
for /opt/ltsp/i386 (/): no export entry
```

Tabla 4-16 Muestra el contenido del archivo /var/log/messages

Este mensaje se puede interpretar fácilmente y confirma que efectivamente el directorio que la terminal pretende montar, no está siendo exportado por el servidor.

Para solucionar esto, basta con verificar el contenido del archivo /etc/exports y añadir si es que no existe, la siguiente línea:

```
/opt/ltsp/i386     192.168.0.0/255.255.255.0(ro,no_root_squash,sync)
```

Tabla 4-17 Muestra el contenido del archivo /

4.3.6 Problemas con el servidor X

La implementación del LTSP puede llegar a ser complicada debido a la gran diversidad de hardware, y un claro ejemplo de esto es la correcta configuración del servidor X.

Si se está utilizando una tarjeta de video y un monitor relativamente modernos, es casi seguro que estén soportados por el servidor XFree86 y por lo tanto no habrá mucho problema en que funcionen adecuadamente; sin embargo cuando se trata de equipos antiguos es necesario realizar algunos ajustes adicionales al servidor.

Los scripts de inicialización del LTSP tratarán de ejecutar el servidor X en las terminales, para que éste se comunique con el Display Manager del servidor. Si por alguna razón el servidor X no puede ser iniciado al primer intento, se probará hacerlo en 10 ocasiones más hasta que finalmente se muestre un mensaje de error si no se tuvo éxito en la conexión.

Cuando sucede esto, lo primero que se debe intentar es cambiar el servidor X especificado en el archivo lts.conf; por default se tendrá configurado el servidor XFree86 pero si la tarjeta de video es algo antigua es muy probable que se deba usar otro servidor X, por ejemplo Xvesa.

Antes de poder modificar el archivo lts.conf, será necesario instalar el paquete correspondiente al servidor Xvesa; el cual puede obtenerse en la página oficial del LTSP.

Para instalar el paquete se tiene que teclear el siguiente comando:

```
rpm –ivh  ltsp_x_vesa-3.0-1.k12ltsp.2.3.0.i386.rpm
```

Este paquete instalará los componentes necesarios para que el servidor Xvesa esté disponible para las terminales, para ello se creará el archivo Xvesa en la ruta:

```
/opt/ltsp/i386/usr/X11R6/bin/
```

Tabla 4-18 Muestra la ruta del archivo Xvesa

Una vez instalado el paquete, se requiere modificar el parámetro XSERVER en la sección de la terminal con problemas; se tendrá que cambiar el valor Xfree86 por Xvesa.

```
XSERVER = Xvesa
```

Después de efectuar los cambios correspondientes, se debe reiniciar la terminal para verificar si el servidor Xvesa soporta su tarjeta de video y monitor.

En caso de que el servidor Xvesa tampoco funcione, se tendrá que recurrir a otras alternativas, por ejemplo los servidores vga o svga; por lo cual es necesario repetir el proceso que se utilizó con Xvesa.

Hay ocasiones en que el servidor X si soporta a la tarjeta de video de la terminal, sin embargo la configuración del monitor no es la apropiada.

En modelos de monitores recientes, la configuración automática es suficiente, pero en monitores antiguos es necesario especificar algunos parámetros adicionales; los cuales definen la resolución así como la frecuencia de refresco de la pantalla.

Se tendrá que agregar los siguientes parámetros al archivo lts.conf, en la sección de la terminal:

```
X_HORZSYNC          = "30.9-49.0"
X_VERTREFRESH  = "50.0-100.0"
X_MODE_0          = 800x600
```

O bien se podrá especificar un modeline en el parámetro X_MODE_0, por ejemplo:

```
X_HORZSYNC  = "30.9-49.0"
X_VERTREFRESH = "50.0-100.0"
X_MODE_0 = 800x600  40     800  840  968 1056   600 601 605 628
    +hsync +vsync
```

Con las modificaciones antes descritas es muy probable que el servidor X inicie correctamente en cada una de las terminales, pero aun es necesario que éste logre establecer una conexión con el Display Manager que corre en el servidor.

4.3.7 Problemas con el Display Manager

El Display Manager (Administrador de Sesiones Gráficas) es un demonio que corre en el servidor, el cual espera que un servidor X se ponga en contacto con él. En el momento en que se establece la conexión se mostrará una caja de diálogo de inicio de sesión, para que el usuario pueda realmente acceder a las aplicaciones del servidor.

Los tres Display Managers más comunes en Linux son:

XDM - Forma parte del sistema estándar de X Window.
GDM - El "Gnome Display Manager" forma parte del entorno de escritorio Gnome.
KDM - El "KDE Display Manager" forma parte del entorno de escritorio KDE.

En ocasiones el servidor X logra iniciar correctamente en la terminal, pero no puede establecer una conexión con el Display Manager; cuando sucede esto, se presenta una pantalla gris con un cursor en forma de X en lugar de la caja de diálogo de inicio de sesión. Las posibles razones son las siguientes:

4.3.7.1 El Display Manager no está en ejecución

Para determinar si algún Display Manager está corriendo en el servidor se puede utilizar el siguiente comando:

```
netstat -ap | grep xdmcp
```

La salida de éste comando mostrará si hay algún proceso escuchando en el puerto xdmcp (177).

```
udp 0 0 *:xdmcp *:* 1255/gdm
```

Tabla 4-19 Muestra la salida para determinar si algún display manager esta activo

En este ejemplo se puede observar que gdm está corriendo con el PID 1255 y está escuchando en el puerto de XDMCP. En caso de que no se muestre salida alguna, significará que ningún Display Manager está corriendo; por lo cual será necesario reiniciar el sistema para reactivar el Display Manager por defecto que utiliza el servidor.

4.3.7.2 El Display Manager ignora peticiones de máquinas remotas

Si se ha comprobado que algún Display Manager está corriendo, entonces es posible que haya sido configurado para ignorar peticiones XDMCP desde máquinas remotas; para solucionar esto, es necesario verificar los archivos de configuración correspondientes al Display Manager utilizado.

4.3.7.2.1 XDM

En el archivo xdm-config la siguiente línea deberá estar comentada para que se admitan peticiones remotas:

```
#    DisplayManager.requestPort: 0
```

Y el archivo Xaccess deberá contener una línea que comience con un asterisco (*):

```
*    # Aceptar peticiones remotas
```

Ambos archivos generalmente se encuentran en la ruta /etc/X11/xdm/ pero si no es así, se pueden buscar con el comando locate.

4.3.7.2.2 KDM

El archivo kdmrc contiene una sección con la etiqueta [xdmcp], dentro de la cual hay un parámetro que deberá estar configurado de la siguiente manera:

```
Enable = true    #true habilita las peticiones remotas
```

Este archivo generalmente se encuentra en la ruta /etc/kde/kdm/ pero si no es así, se puede buscar con el comando locate.

4.3.7.2.3 GDM

El archivo gdm.conf contiene una sección con la etiqueta [xdmcp], dentro de la cual hay un parámetro que deberá estar configurado de la siguiente manera:

```
Enable = true    #true habilita las peticiones remotas
```

Este archivo generalmente se encuentra en la ruta /etc/X11/gdm/ pero si no es así, se puede buscar con el comando locate.

4.3.7.3 Conflicto del Display Manager con el DNS

Si el Display Manager definitivamente está en ejecución y está escuchando las peticiones de máquinas remotas, entonces el problema puede deberse a un simple error en el mapeo de direcciones IP a nombres, por lo que el Display Manager no sabrá con que terminal establecer la comunicación.

Para solucionar este problema es necesario configurar el archivo hosts, agregando una entrada para cada una de las terminales, por ejemplo:

```
192.168.0.1   ws001    ws001.localdomain   ws001
192.168.0.2   ws002    ws002.localdomain   ws002
```

4.3.8 Archivos de sistema

Finalmente, después de resolver los problemas detectados y realizar las modificaciones correspondientes a los archivos de configuración estos quedarán así:

```
# Archivo de configuración dhcpd.conf
ddns-update-style            none;
default-lease-time           21600;
max-lease-time               21600;
option subnet-mask           255.255.255.0;
option broadcast-address     192.168.0.255;
option routers               192.168.0.254;
option domain-name-servers   192.168.0.254;
option domain-name           "yourdomain.com";
option root-path             "192.168.0.254:/opt/ltsp/i386";
option option-128 code 128 = string;
option option-129 code 129 = text;

shared-network WORKSTATIONS {
   subnet 192.168.0.0 netmask 255.255.255.0 {
   }
}

group    {
   use-host-decl-names      on;
   option log-servers       192.168.0.254;
   host ws001 {
      hardware ethernet     00:60:8C:CA:BC:FF;
      fixed-address         192.168.0.1;
      filename              "/lts/vmlinuz-2.4.19-ltsp-1";
      option option-128     e4:45:74:68:00:00;
      option option-129     "NIC=3c509";
   }
   host ws002 {
      hardware ethernet     00:A0:24:24:F6:0F;
      fixed-address         192.168.0.2;
      filename              "/lts/vmlinuz-2.4.19-ltsp-1";
      option option-128     e4:45:74:68:00:00;
      option option-129     "NIC=3c509";
   }
   host ws003 {
      hardware ethernet     00:80:AD:3B:8F:E4;
      fixed-address         192.168.0.3;
      filename              "/lts/vmlinuz-2.4.19-ltsp-1";
   }
}
```

Tabla 4-20 Muestra le archivo completo de configuración exports

```
# Archivo de configuración exports

/opt/ltsp/i386                  192.168.0.0/255.255.255.0(ro,no_root_squash,sync)
/var/opt/ltsp/swapfiles   192.168.0.0/255.255.255.0(rw,no_root_squash,async)
```

Tabla 4-21 Muestra el archivo completo de configuración exports

```
# Archivo de configuración hosts.allow

bootpd:   0.0.0.0
in.tftpd:  192.168.0.
portmap:   192.168.0.
mountd:    192.168.0.
ALL:        127.0.0.1
ALL:      192.168.0.254
ALL:      192.168.0.1
ALL:      192.168.0.2
ALL:      192.168.0.3
```

Tabla 4-22 Muestra el archivo completo de configuración hosts.sllow

```
# Archivo de configuración lts.conf

[Default]
      SERVER          = 192.168.0.254
      XSERVER         = auto
      X_MOUSE_PROTOCOL  = "PS/2"
      X_MOUSE_DEVICE    = "/dev/psaux"
      X_MOUSE_BUTTONS   = 3
      X_USBMOUSE_PROTOCOL= "IMPS/2"
      X_USBMOUSE_DEVICE  = "/dev/input/mice"
      X_USBMOUSE_BUTTONS = 5

      XkbSymboles      = "us(pc101)"
      XkbModel         = "pc101"
      XkbLayout        = "us"

      USE_XFS          = N
      LOCAL_APPS       = N
      RUNLEVEL         = 2

      SOUND            = Y

      VOLUME           = 75

[ws001]
      XSERVER          = Xvesa
      LOCAL_APPS       = N
      USE_NFS_SWAP     = N
      SWAPFILE_SIZE    = 48m
      RUNLEVEL         = 5
      X_HORZSYNC       = "30.9-49.0"
      X_VERTREFRESH    = "50.0-100.0"
      X_MODE_0         = 800x600
[ws002]
      XSERVER          = Xvesa
      LOCAL_APPS       = N
```

```
    USE_NFS_SWAP      = N
    SWAPFILE_SIZE     = 48m
    RUNLEVEL          = 5
    X_HORZSYNC        = "30.9-49.0"
    X_VERTREFRESH     = "50.0-100.0"
    X_MODE_0          = 800x600
[ws003]
    XSERVER           = Xvesa
    LOCAL_APPS        = N
    USE_NFS_SWAP      = N
    SWAPFILE_SIZE     = 48m
    RUNLEVEL          = 5
    X_MODE_0          = 800x600
```

Tabla 4-23 Muestra el archivo completo de configuración lts.conf

4.4 Conexión en red

De acuerdo con la metodología propuesta, cuando se logra dar solución a los problemas detectados y todas las terminales inician una sesión gráfica individualmente; el siguiente paso es probar su funcionamiento conectándolas en red.

Este proceso es relativamente sencillo, pero hay que considerar algunos aspectos importantes en su implementación, los cuales básicamente son:

- Uso de cable UTP directo en lugar de cable cruzado
- Uso de un switch en lugar de un hub para garantizar un mejor rendimiento en la red
- Asignación correcta de direcciones IP para evitar conflictos entre las terminales

Tomando en cuenta los puntos anteriores, los pasos para implementar la red son los siguientes:

1. Conectar el servidor al primer puerto del switch mediante un cable directo
2. Conectar la terminal 1 al segundo puerto del switch, la segunda terminal al tercer puerto y así sucesivamente, todas estas conexiones mediante cable directo
3. Encender el servidor e iniciar los servicios necesarios del LTSP
4. Encender el switch
5. Encender cada una de las terminales y autenticarse para iniciar una sesión remota en el servidor

El siguiente diagrama muestra la configuración de los equipos al estar conectados en red:

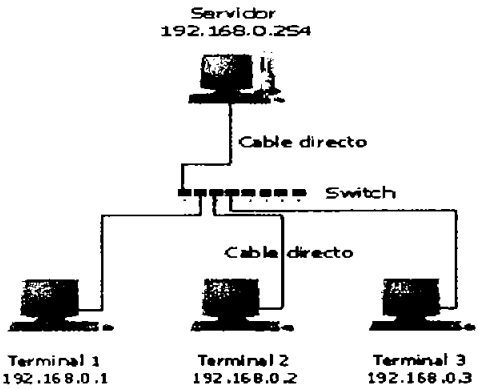


Figura 4-2 Diagrama que muestra la configuración de los equipos en red

La siguiente figura muestra a las terminales con la pantalla de bienvenida del LTSP al estar en red.

Figura 4-3 Imagen de las terminales en red con la pantalla de bienvenida del LTSP

4.5 Conexión a internet

Inicialmente, la red de terminales gráficas ha sido configurada para funcionar de manera local, esto de por si ya ofrece muchas ventajas, las cuales han sido mencionadas a lo largo de este trabajo, pero debido a que actualmente el acceso a Internet es indispensable, se optó por configurar la red local para que tenga acceso a éste.

Para establecer una conexión a Internet, debe existir un medio por el cual la red local se pueda comunicar con las redes externas; esto básicamente se logra con una tarjeta de red adicional en el servidor, lo que permite que éste también tenga la función de un gateway. Aunque es posible utilizar una sola tarjeta de red creando un "alias", esto no es recomendable debido al tráfico y a la carga que se generaría en dicho dispositivo.

Al configurar el servidor para que tenga salida a Internet por medio de la segunda tarjeta de red (eth1), automáticamente todas las terminales también tendrán acceso. Es necesario entender que en una red implementada con el LTSP, cualquier usuario conectado en un ambiente grafico, es en realidad un usuario local del servidor, por lo tanto, cualquier aplicación que lo requiera puede usar la misma conexión a Internet del servidor sin ningún inconveniente.

El programa redhat-config-network permite configurar de forma sencilla los dispositivos de red instalados en el sistema, éste puede ser ejecutado desde una terminal con el siguiente comando:

```
redhat-config-network &
```

Este programa detecta de forma automática los dispositivos de red instalados y los muestra en la sección "Dispositivos", en este caso deben aparecer dos dispositivos: eth0 y eth1.

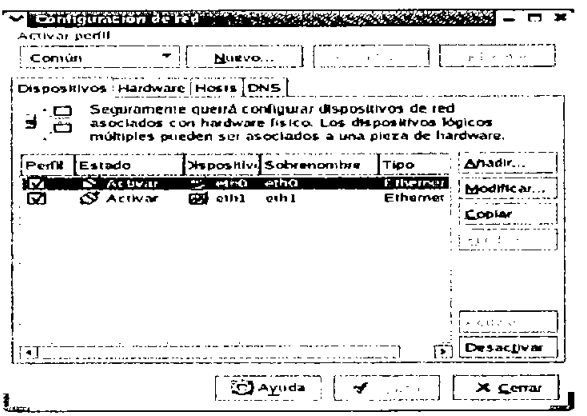


Figura 4-4Pantalla donde se muestran los dispositivos lógicos de red

En la pestaña Hardware, se debe verificar que se haya detectado correctamente la marca y modelo de las tarjetas de red; esto es necesario debido a que pueden ocurrir errores en los controladores, si estos no son los apropiados.

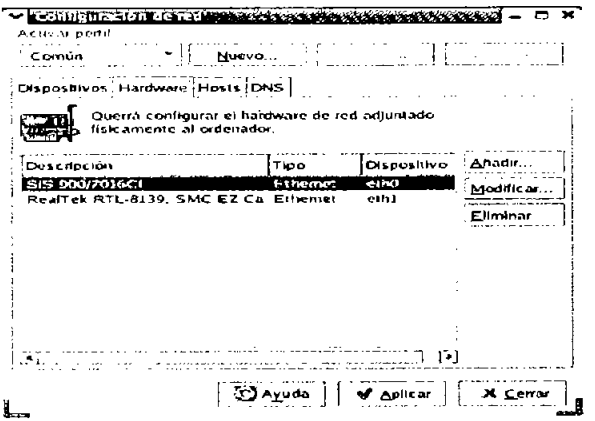


Figura 4-5 Pantalla donde se muestran los dispositivos físicos de red

La tarjeta eth0 permitirá comunicar al servidor con la red de terminales gráficas, para configurarla se debe regresar a la pestaña "Dispositivos" y seleccionar el dispositivo eth0; hacer click en "Modificar".

En la siguiente pantalla, se tiene que desactivar la opción "Obtener las configuraciones de direcciones IP automáticamente con dhcp", se activará la opción "Configurar las direcciones IP de manera estática". Se debe ingresar la dirección IP 192.168.0.254, la máscara de subred 255.255.255.0 y la puerta de enlace 192.168.0.254. Hacer click en "OK" para continuar.

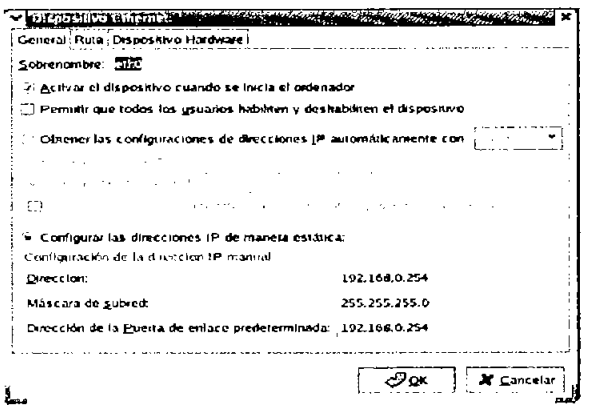


Figura 4-6 Configuración del dispositivo eth0

La tarjeta eth1 permite comunicar al servidor con las redes externas, en realidad es en este dispositivo donde se configura el acceso a Internet, por lo que se debe seleccionar y hacer click en "Modificar". En la siguiente pantalla, se tiene que desactivar la opción "Obtener las configuraciones de direcciones IP automáticamente con dhcp", se activará la opción "Configurar las direcciones IP de manera estática". Se debe ingresar la dirección IP 132.248.173.84, la máscara de subred 255.255.255.0 y la puerta de enlace 132.248.173.254. Hacer click en "OK" para continuar.

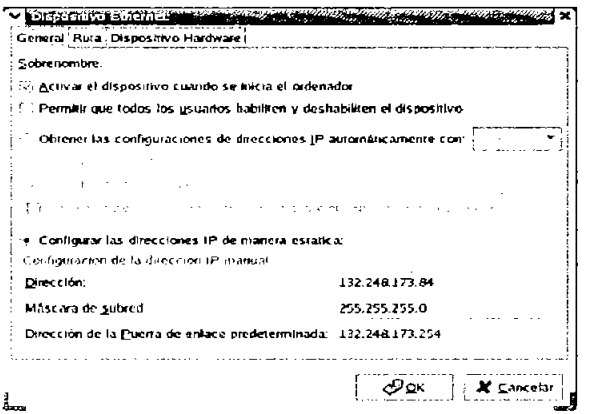


Figura 4-7 Configuración del dispositivo eth1

En la pestaña "DNS", se debe ingresar el nombre de host "Electromartin", el DNS primario 132.248.204.1 y el DNS secundario 132.248.10.2. En la sección "Ruta de búsqueda DNS", se debe añadir el dominio "aragon.unam.mx". Una vez ingresados los datos, hacer click en "Aplicar" para confirmar los cambios.

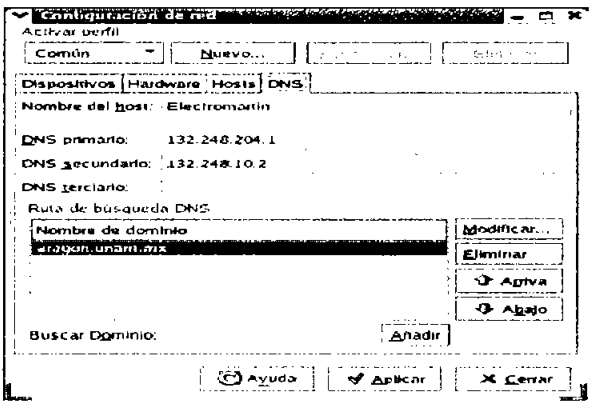


Figura 4-8 Configuración del DNS

4.6 Software específico

El LTSP es un claro ejemplo de las ventajas y posibilidades que el software libre puede ofrecer, ya que mediante la adecuada combinación de diferentes servicios y programas se logró implementar la red de terminales gráficas en el laboratorio de electrónica.

Como ya se había mencionado, dicha red proporciona entre otras cosas, un sistema centralizado de almacenamiento de información y la posibilidad de utilizar una gran variedad de aplicaciones, la mayoría de ellas distribuidas bajo la licencia GNU/GPL.

Sin embargo, no es conveniente instalar de una forma indiscriminada programas que tal vez nunca serán utilizados; para evitar esto se pensó en la necesidad de los alumnos de contar con software relacionado con la electrónica, el cual puede clasificarse en tres principales grupos: diseño, simulación y programación de circuitos.

Actualmente ya se cuenta con software de este tipo, por ejemplo: el Electronics Workbench, Pspice, Warp, Mplab, entre otros; pero debido a que estos programas corren en Windows se optó por buscar software propio de Linux que pudiera ser útil en el laboratorio y que tuviera características semejantes a los anteriores.

Cabe mencionar que el software para Windows podrá seguir utilizándose debido a que las terminales han sido configuradas para que puedan seguir arrancando este sistema operativo si así se requiere. En Internet hay muchos programas de electrónica disponibles, pero solo se describirán los que se consideraron más funcionales y amigables para el usuario.

4.6.1 Diseño

Este tipo de software es utilizado para crear diagramas o esquemas que representan gráficamente un circuito electrónico, éstos pueden imprimirse o exportarse a otros programas mediante la generación de un archivo compatible. Los programas de diseño no tienen la capacidad de simular o interpretar el funcionamiento del diagrama, sin embargo son muy útiles debido a que permiten generar fácilmente la estructura de casi cualquier circuito eléctrico - electrónico.

4.6.1.1 Eagle

Es un software de diseño de esquemas PCB (Printed Circuit Boards) con motor de autotrazado y una gran variedad de componentes. Es comercial, pero existe una licencia freeware (educacional) que limita el tamaño máximo de la placa de circuito impreso.

El programa consta de tres módulos principales:

- Editor de circuito impreso (Layout Editor)
- Editor de diagramas esquemáticos (Schematic Editor)
- Organizador de pistas (Autorouter)

Estos tres módulos están integrados en una sola interfaz de usuario, lo que permite crear fácilmente el diseño para la placa, a partir del circuito implementado en el editor esquemático.

La versión de evaluación del software permite crear circuitos impresos de hasta dos caras y con área máxima de 100 x 80 mm, suficiente para una gran variedad de circuitos sencillos.

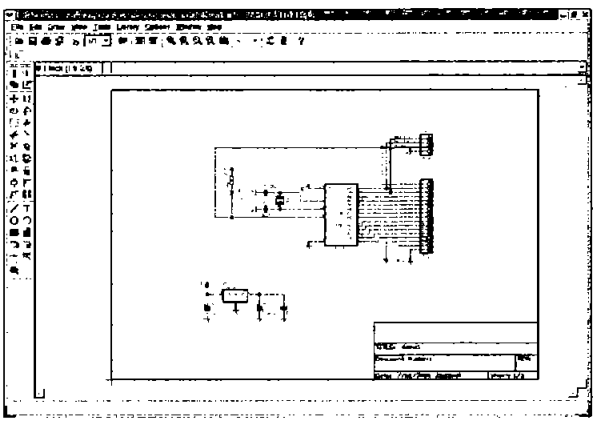


Figura 4-11 Imagen del programa Eagle

Para la instalación de Eagle, se debe descargar la última versión del código fuente[16] y seguir los pasos descritos en el capítulo 2 (métodos de instalación).

4.6.1.2 Xcircuit

Xcircuit es un programa que permite dibujar de una manera sencilla, diagramas de circuitos eléctricos de gran calidad, los cuales pueden ser exportados a otros programas como el Spice. También es posible realizar esquemas de diversas temáticas, debido a que pueden utilizarse bibliotecas de distintos objetos.

[16] Página para descargar el código http://www.cadsoft.de/info.htm

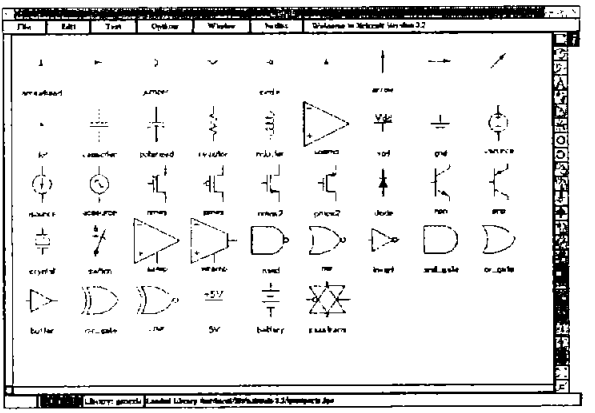


Figura 4-12 Imagen del programa Xcircuit

Para la instalación de Xcircuit, se debe descargar la última versión del código fuente[17] y seguir los pasos descritos en el capítulo 2 (métodos de instalación).

4.6.2 Simulación

Estos programas también permiten crear diagramas de circuitos electrónicos, pero tienen la gran ventaja de que se puede analizar el comportamiento físico del circuito; se puede determinar el voltaje, corriente o los estados lógicos por los que va pasando cada componente. Por lo general, los simuladores cuentan con varias herramientas entre las cuales se encuentran: los diagramas de componentes, diagramas de onda y cálculo de parámetros.

4.6.2.1 Klogic

Klogic es una herramienta para construir y simular circuitos digitales que contienen componentes estándar como AND, OR, XOR, Flip Flops RS y JK. Si se requiere construir circuitos más complejos y reutilizables se tiene la opción de diseñar subcircuitos.

Mientras se está construyendo el circuito, la simulación está ejecutándose de forma permanente, además para pruebas extendidas el programa tiene la capacidad de simular el circuito paso a paso para reconocer los estados por los que éste va pasando. También es posible desplegar la señal de flujo de los componentes de un circuito de forma gráfica e intuitiva.

[17] Página para descargar el código http://bach.ece.jhu.edu/~tim/programs/xcircuit/welcome.html

paquetes; lo que permite que su uso sea fácilmente extendido para nuevas áreas de aplicación.

Los principales componentes de Ksimus son:

- Compuertas AND, NAND, OR, NOR, XOR, XNOR
- Flip Flops RS, JK y JK-MS
- Generador de pulsos
- LED
- Latches D
- Contador
- Display de 7 segmentos
- Multiplexor y demultiplexor
- Generador de ondas
- Aritméticos (add, sub, mul, div, mod, abs)
- Trigonométricos (sin, asin, cos, acos, tan, atan, atan2)
- Hiperbólicos (sinh, asinh, cosh, acosh, tanh, atanh)
- Exponenciales (log, log10, exp, pow, square)

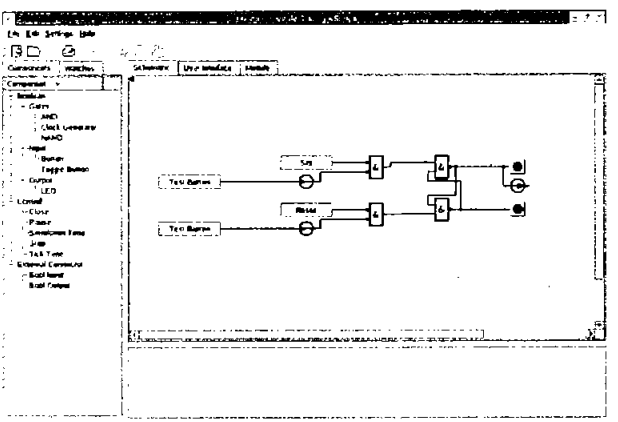


Figura 4-15 Imagen del programa Ksimus

Para la instalación de Ksimus, se debe descargar la última versión del código fuente[19] y seguir los pasos descritos en el capítulo 2 (métodos de instalación).

[19] Página para bajar el código fuente http://ksimus.berlios.de/index.html

4.6.2.3 Tkgate

Tkgate es un simulador de circuitos digitales, basado en un editor gráfico tcl/tk. Soporta una amplia gama de componentes básicos, pero además se tiene la capacidad de construir módulos mediante el diseño jerárquico de circuitos.

Los principales componentes de Tkgate son:

- Compuertas AND, NAND, OR, NOR, XOR, XNOR
- Transistores NMOS y PMOS
- Buffers triestado
- Componentes ALU (sumadores y multiplicadores)
- Elementos de memoria (registros, RAMs, ROMs)

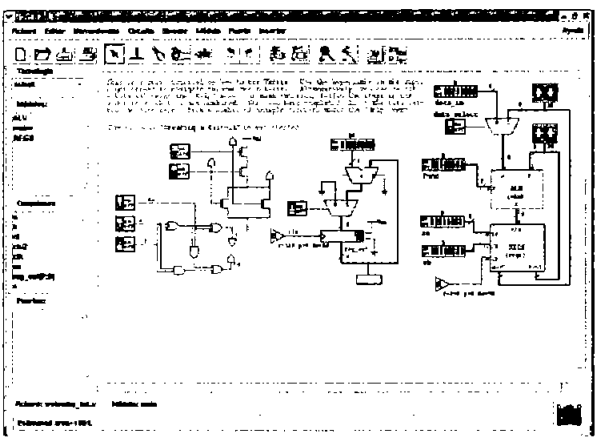


Figura 4-16 Imagen del programa Tkgate

Para la instalación de Tkgate, se debe descargar la última versión del código fuente[20] y seguir los pasos descritos en el capítulo 2 (métodos de instalación).

4.6.2.4 Vipec

Vipec es una poderosa herramienta para el análisis de circuitos de alta frecuencia y redes eléctricas lineales. Es adecuado para simular en el dominio de la frecuencia, circuitos relacionados con las comunicaciones, como: amplificadores, filtros y líneas de transmisión.

[20] Página para bajar el código fuente http://www.tkgate.org

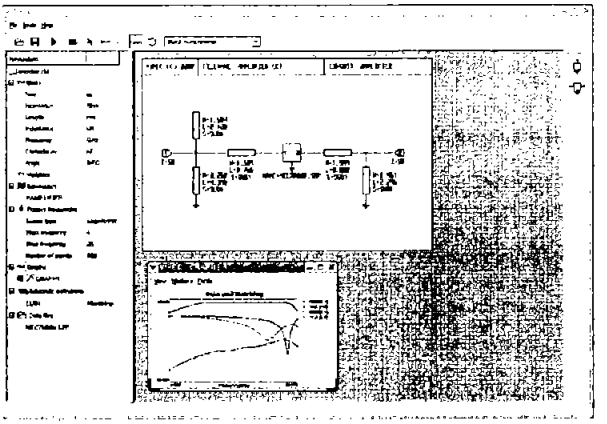


Figura 4-17 Imagen del programa Vipec

Para la instalación de Vipec, se debe descargar la última versión del código fuente[21] y seguir los pasos descritos en el capítulo 2 (métodos de instalación).

4.6.2.5 Gpsim

Este entorno de simulación para PIC's soporta microprocesadores de 12, 14 y 16 bits, además tiene la capacidad de simular en tiempo real sistemas con un ciclo de reloj de 20 MHz.

GPSIM cuenta con un depurador, un desensamblador, un manejador de estímulos, un mapa de memoria y algunos plugins que permiten simular componentes externos al PIC como una pantalla LCD.

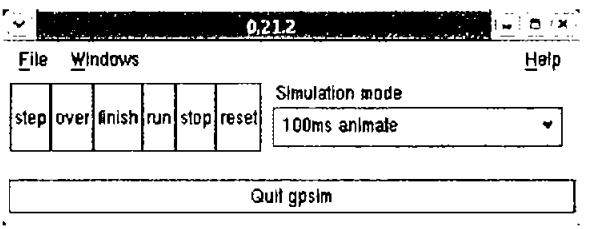


Figura 4-18 Imagen del programa Gpsim

[21] Página para bajar el código fuente http://vipec.sourceforge.net

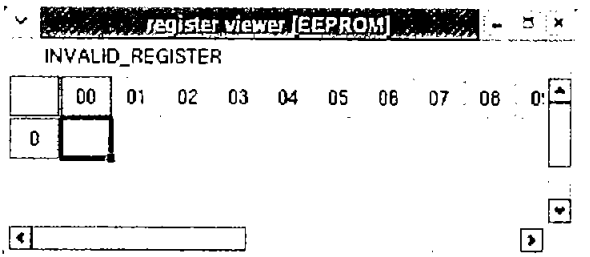


Figura 4-19 Imagen de un ejemplo de programación del programa Gpsim

Para la instalación de Gpsim, se debe descargar la última versión del código fuente[22] y seguir los pasos descritos en el capítulo 2 (métodos de instalación).

4.6.3 Programación

Estas aplicaciones permiten diseñar mediante lenguajes de alto nivel, la estructura lógica de circuitos programables, por ejemplo: memorias, microcontroladores, microprocesadores, PAL, GAL, etc.

Una vez que el diseño es terminado, éste tiene que ser copiado en el chip correspondiente, lo cual es posible utilizando un dispositivo especial denominado "grabador". En ocasiones es posible simular el comportamiento del circuito antes de ser grabado, lo que permite la optimización y depuración del diseño.

4.6.3.1 BlueHDL

BlueHDL es una suite de herramientas VHDL, que consiste en un editor y compilador VHDL, un motor de simulación, y una interfaz de visualización de ondas.

Permite diseñar con lenguajes de alto nivel logrando desarrollar circuitos programables, es una buena opción ya que para todo circuito por sencillo que sea se busca realizarlo programando un microcontrolador.

[22] Página para bajar el código fuente http://www.dattalo.com/gnupic/gpsim.html

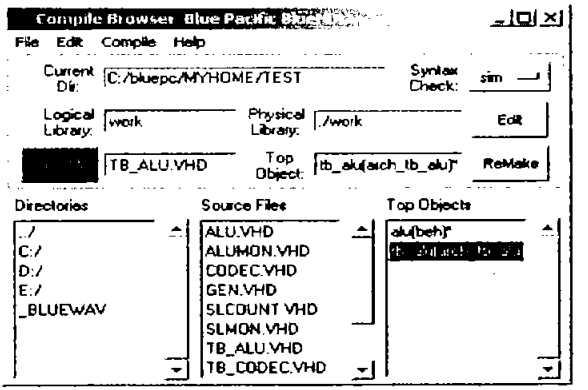


Figura 4-20 Imagen del programa BlueHDL

Para la instalación de BlueHDL, se debe descargar la última versión del código fuente[23] y seguir los pasos descritos en el capítulo 2 (métodos de instalación).

Con todas las instalaciones y configuraciones previas se comprobó que efectivamente cada una de las terminales inició una sesión gráfica adecuadamente, apoyándonos en la parte de detección y solución de problemas; además la red y la conexión a internet funcionó tal y como se esperaba. Ahora analizaremos el rendimiento del servidor para conocer cuántas terminales realmente pueden funcionar adecuadamente.

[23] Página para bajar el código fuente http://www.bluepc.com/index.html

PRUEBAS Y ANÁLISIS DE LOS RESULTADOS

OBJETIVO ESPECÍFICO

En primer lugar se comprobará el rendimiento del servidor con sus tres terminales ejecutando diversas aplicaciones, también se analizarán algunas medidas y acciones que garanticen un mínimo de seguridad y funcionalidad en la red; y por último se darán las pautas para retomar posibles proyectos relacionados o simplemente para mejorar los servicios que proporciona el servidor.

5

INTRODUCCIÓN

Para el desarrollo de este apartado se planteó la necesidad de analizar el rendimiento del servidor para tres terminales, la idea básica es dar a conocer el porcentaje de recursos que requieren las terminales para funcionar apropiadamente sin colapsarse o alentar al servidor. Los valores se obtuvieron con la ayuda del comando TOP y en el caso del procesador sus valores son dados en porcentajes para una mejor comprensión.

Por otro lado se puntualizan las ventajas de utilizar el sistema operativo Linux que proporciona algunas características con respecto a la administración y seguridad del sistema, no es necesario realizar configuraciones complejas para garantizar la seguridad que se desea, ya que el propio sistema viene con valores predefinidos que proporcionan un nivel aceptable de seguridad; pero no está de más tomar ciertas medidas y precauciones las cuales se darán en dicho apartado.

Con respecto a la administración, Linux es un sistema realmente amigable y potente, facilita el manejo de usuarios y lo más importante es que mantiene espacios restringidos para un mejor control del sistema. Es importante mencionar que la administración y la seguridad están íntimamente ligados ya que sino hubiera un control estricto del espacio de usuarios, ésto podría ser una causa para que se realizara un ataque al sistema comprometiendo la seguridad del mismo.

Una vez que se ha comprobado que el sistema es realmente estable y se tiene un control de los archivos de sistema, así como de los recursos que se le brindan a los usuarios para no permitirles un uso inapropiado de los mismos; se puede garantizar que se cuenta con una red estable, fácil de administrar y con un nivel aceptable de seguridad.

Pero eso no es todo, ya que es posible hacer mejoras o innovaciones; hay varias aplicaciones u opciones que se pueden implementar para mejorar los servicios brindados en la red de terminales gráficas.

5.1 Pruebas de rendimiento

En este apartado se probará el rendimiento del servidor y para esto se tendrá que encender terminal por terminal para tener la oportunidad de ir monitoreando las peticiones de recursos que se hacen al servidor, y a su vez tener una relación de la cantidad de memoria que se requiere para soportar n cantidad de terminales funcionando adecuadamente.

Estas pruebas ayudarán a comprobar que Linux administra eficientemente los recursos disponibles, principalmente de memoria y de procesamiento del sistema, los valores de las tablas muestran claramente cuánto se consume de los recursos del servidor pero a la vez es posible visualizar que se llega a un estado de estabilidad que impide que el sistema deje de funcionar.

El servidor cuenta con las siguientes características:

Procesador	700 Mhz
Memoria	376640 Kb
Swap	786432 Kb

Se recurrió al siguiente comando para obtener la información del uso de procesador, memoria y swap del servidor:

```
top
```

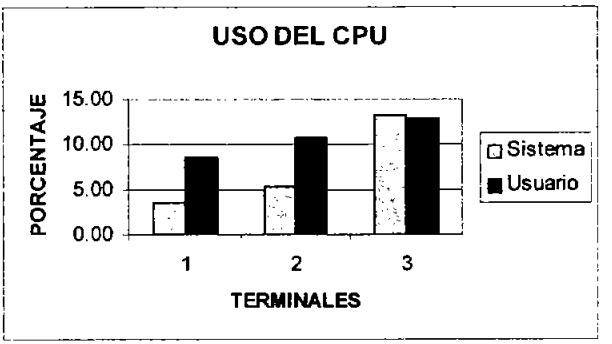
Los resultados obtenidos al momento de encender las tres terminales, se muestran en la siguiente tabla:

TERMINAL	CPU (%)		MEM (Kb)	SWAP (Kb)
	SISTEMA	USUARIO		
1	3.50	8.50	348868	0
2	5.30	10.80	364496	0
3	13.30	12.90	371024	0

Tabla 5-1 Resultados obtenidos al momento de encender las tres terminales

En la siguiente gráfica se puede notar que el procesador no tiene gran desgaste al arrancar las terminales y esto sin tomar en cuenta lo que requiere el servidor. Se puede ver que con la primer terminal se utilizó un 8.5% entonces con una segunda debería de consumir un 17% y a su vez con una tercera gastaría un 25.5% de la capacidad del procesador.

Pero esto no es así ya que realmente sólo se empleo un 13% aproximadamente y no el 25% que se tenía estimado, este comportamiento se debe principalmente a que el kernel de Linux administra eficientemente el uso del procesador; el planificador de procesos es el encargado de gestionar el uso de este recurso ya que verifica la prioridad de cada uno de los procesos y los manda a ejecución solo cuando es necesario, optimizando en gran medida los tiempos del procesador.



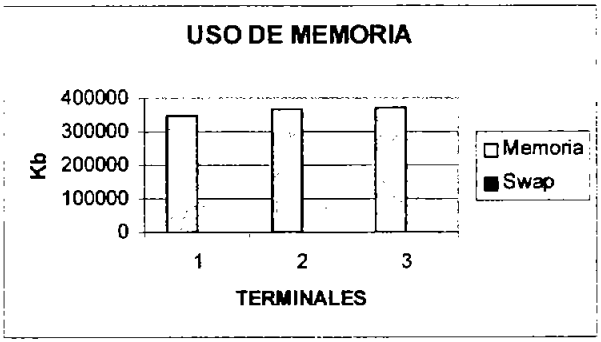
Gráfica 5-1 El uso del CPU

Se puede observar también que la memoria se está utilizando prácticamente en su totalidad, a pesar de que solo es el arranque y aún no se ha ejecutado alguna aplicación en las terminales; cabe mencionar que el sistema aun no ha requerido el uso de la memoria swap.

Bajo estas circunstancias se podría pensar que el sistema esta desperdiciando la memoria, pero no es así, ya que Linux utiliza toda la memoria libre para caché de una forma más dinámica. En un estado estable prácticamente toda la memoria principal está ocupada con fragmentos de procesos, por lo que se tendrá acceso directo a la mayor cantidad de procesos posibles disminuyendo el tiempo en la ejecución de un programa.

El kernel de Linux soporta ejecutables cargados por demanda de páginas, donde solo aquellos segmentos de un programa que se requieren son leídos del disco y puestos en la memoria.

Así, si varias instancias de un programa son ejecutadas simultáneamente, sólo una copia del código del programa estará en memoria y por esta razón el sistema se mantiene estable al ejecutar los mismos programas en las terminales.



Gráfica 5-2 El uso de la memoria

Una vez que las terminales han iniciado una sesión gráfica, es necesario verificar el rendimiento del servidor pero en este caso corriendo aplicaciones del usuario. Para esto, se eligieron 3 aplicaciones que se consideraron como las más utilizadas por los alumnos (Navegador Web, procesador de textos y hoja de cálculo), así como 3 programas específicos para el diseño y simulación de circuitos electrónicos (Xcircuit, Tkgate y Bluehdl).

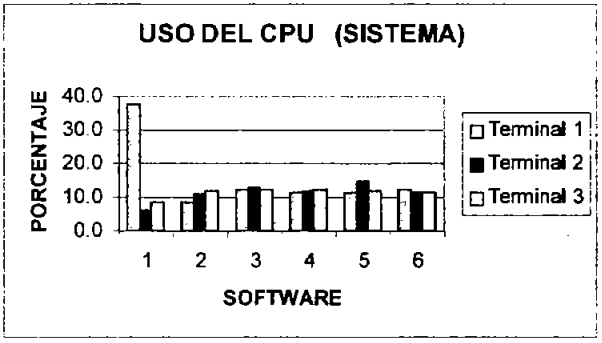
La prueba consistió en ejecutar una aplicación en la primer terminal, en este caso el navegador, luego se anotaron los valores de los recursos actuales (procesador, memoria y swap); posteriormente se hizo lo mismo con la segunda y tercer terminal. Después que se hicieron las mediciones correspondientes al primer programa, se prosiguió a ejecutar el procesador de texto, continuando con la misma mecánica se ejecutaron las aplicaciones restantes; hasta que al final se recopiló la información de cuanto fue el consumo de recursos por las tres terminales y a su vez para las 6 aplicaciones.

La siguiente tabla muestra los datos arrojados en la prueba antes descrita:

SOFTWARE	CPU (%)						MEM (Kb)			SWAP (Kb)		
	Term 1		Term 2		Term 3		Term 1	Term 2	Term 3	Term 1	Term 2	Term 3
	SIST	USR	SIST	USR	SIST	USR						
Navegador	37.6	20.5	6.0	15.9	8.7	19.7	371040	371120	365912	0	0	0
Proc. de texto	8.7	19.7	11.0	12.2	12.1	13.6	372036	371980	372216	0	0	0
Hoja de cálc.	12.2	15.2	12.9	11.6	12.5	14.7	372200	371180	371464	0	0	1132
Xcircuit	11.5	15.0	12.1	13.6	12.5	19.1	372152	372092	370964	1132	1132	1104
Tkgate	11.4	14.5	14.6	14.8	11.7	19.6	371568	371520	371508	1104	1104	1104
Bluehdl	12.4	14.8	11.4	16.8	11.2	16.1	371412	371792	371712	1104	1104	1104

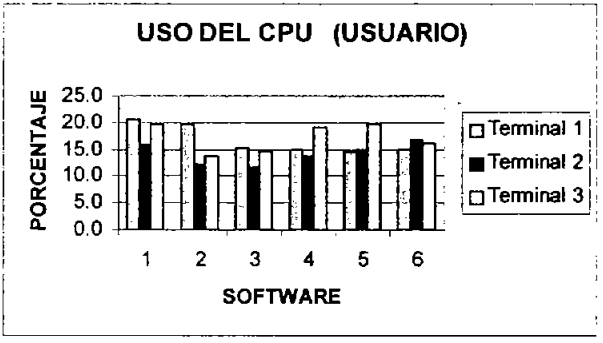
Tabla 5-2 Resultados obtenidos de las pruebas realizadas

Como puede observarse, al ejecutar el navegador en la primer terminal, se obtuvo un alto porcentaje en el uso del procesador; esto se debió a que el sistema le dio prioridad máxima a esta aplicación e incremento la actividad de procesamiento, pero después de cierto tiempo el sistema se volvió a estabilizar gracias a las características del planificador de procesos antes mencionadas.



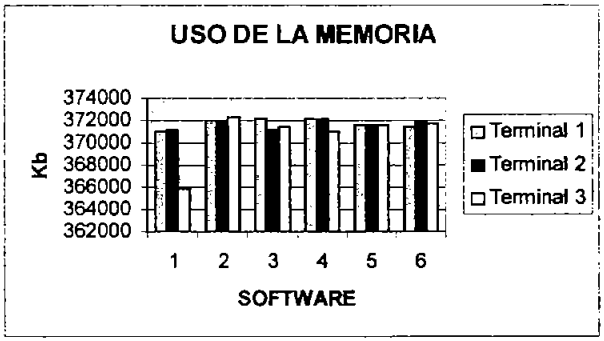
Gráfica 5-3 Uso del CPU (sistema) con las tres terminales

En el caso del uso del procesador por parte de las aplicaciones de usuario, se observa que no hubo mucha variación en el porcentaje; ya que se tiene la ventaja de que se ejecutaron las mismas aplicaciones para las tres terminales, optimizando en gran medida el uso de este dispositivo. Además con estas pruebas también fue posible comprobar la eficiencia de Linux para manejar la concurrencia en las aplicaciones.



Gráfica 5-4 Uso del CPU (usuario) con las tres terminales

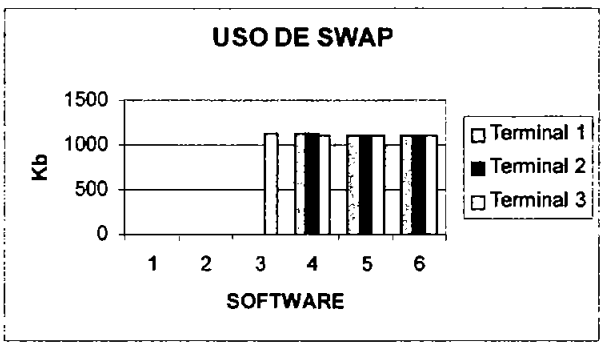
Con respecto a la memoria, se puede notar que al ejecutar cada uno de los programas ésta prácticamente se ha agotado, pero el sistema aún tiene un buen funcionamiento ya que de cierta forma llega a un punto de estabilización que impide que el sistema se colapse.



Gráfica 5-5 Uso de la memoria con las tres terminales

Sin embargo cuando la memoria física se ha terminado, Linux comienza a hacer uso de la memoria swap, sin ella las terminales se hubieran colapsado cuando la tercer terminal hiciera la petición para ejecutar el tercer programa; con esta memoria auxiliar es posible seguir trabajando y tener soporte para ejecutar tres programas más con las tres terminales funcionando adecuadamente.

La memoria swap es importante pero no puede reemplazar a una memoria física, pero puede permitir al sistema ejecutar aplicaciones más grandes guardando en disco duro aquellas partes de código que no se utilizan.



Gráfica 5-6 Uso de swap con las tres terminales

Considerando que se empezó a utilizar la memoria swap a partir del tercer programa y se cuenta con 384 MB de RAM en el servidor, se puede concluir que con un promedio de 3 aplicaciones corriendo, se necesita aproximadamente 100 MB de memoria por cada terminal; de esta manera si en un futuro se requiere expandir la red, se recomienda que por cada terminal que se incorpore, se incremente proporcionalmente el tamaño de la memoria, por ejemplo para 10 terminales se deberá tener aproximadamente 1 GB de RAM.

En cuanto al procesador, se recomienda que para más de 5 terminales, su velocidad sea mayor o igual a 1 GHz, con esto se garantiza un buen desempeño y confiabilidad de la red, solo si se desea utilizar una computadora personal como servidor.

Pero si se desea implementar este proyecto de una manera más formal, es decir, más de 15 terminales, lo más adecuado es invertir en un servidor, ya que es un equipo diseñado para estar conectado las 24 horas del día durante todo el año, el cual tiene que soportar fuertes picos de actividad, y debe tener un alto rendimiento en ejecución de aplicaciones para que los alumnos no vean afectada la velocidad y que cuente básicamente con las siguientes características, tomando como referencia las pruebas que se realizaron.

Procesador: Intel Pentium IV
Frecuencia: 3 Ghz
Memoria: 4GB
Disco Duro: 160 GB

La empresa Hewlett Packard ofrece un servidor económico de $17 000, con estas características realmente no es mucha la inversión tomando en cuenta que de esta forma se podrá tener varias 486 funcionando como estaciones de trabajo, si se invirtiera esta cantidad en computadoras medianamente potentes solo se lograría adquirir aproximadamente 4, sin tomar en cuenta la cantidad faltante por costos de licencias. Qué mejor inversión que la de adquirir este servidor y poner a funcionar esas viejas 486 brindando una potencia aceptable y sin gastos por licencias de software.

5.2 Seguridad

Con el fin de garantizar que la red de terminales gráficas funcione adecuadamente y que los recursos del servidor estén disponibles en cualquier momento, se pretende dar algunas ideas generales de los riesgos existentes y las medidas de seguridad para prevenir los posibles ataques al sistema.

La meta de la mayoría de los ataques a un sistema operativo es conseguir un acceso con privilegios de root, lo que garantiza un control total del sistema; primero se

intentará conseguir acceso como usuario normal e ir incrementando los niveles de privilegio utilizando las posibles vulnerabilidades del sistema.

Se puede decir que con el simple hecho de intentar violar la privacidad de la información, obteniendo privilegios que no le corresponden a un determinado usuario, haciendo un uso desmedido de los recursos o modificando información legítima contenida en una máquina, como puede ser el contenido de una página web o una base de datos; éstas son algunas formas de poner en riesgo el correcto funcionamiento del sistema.

En todo sistema la seguridad simplemente es fundamental, se debe tomar en cuenta que en un ambiente LTSP pueden haber muchos usuarios conectados al servidor al mismo tiempo; por lo que el sistema operativo del servidor tendrá la obligación de proteger a unos usuarios frente a otros y protegerse así mismo.

En primer lugar se debe tener en cuenta que ningún sistema es completamente seguro, solo lo será cuando no este conectado a la red o simplemente este apagado. Desde esta perspectiva se puede partir de que lo único que se puede hacer es aumentar la dificultad para que alguien pueda comprometer la seguridad del sistema.

Por tal motivo, es conveniente establecer algunas políticas que fijen el nivel de seguridad que se requiere, partiendo desde las políticas de administración del servidor, así como algunas políticas de uso para los alumnos, los cuales serán los que potencialmente podrán comprometer la seguridad de la red del laboratorio.

Lo que se desea es garantizar que el sistema permanezca funcionando de forma adecuada con comunicaciones seguras, donde nadie pueda obtener o modificar información a la que no tiene derecho. Una buena planeación ayuda bastante y antes de asegurar el sistema se debe determinar contra qué quiere protegerse y qué riesgos acepta.

Se pueden mencionar algunos requisitos de seguridad:

- Disponibilidad: Mantener la información y los recursos garantizando que no se limite el acceso autorizado a la información y el correcto funcionamiento de los recursos.
- Integridad: Requiere que la información sólo pueda ser modificada por las entidades autorizadas.
- Autenticidad: La información debe permanecer protegida ante falsificaciones, requiere de mecanismos de identificación asegurando que las comunicaciones se realizan entre entidades legítimas.
- Confidencialidad: Pretende evitar la difusión no autorizada de la información, requiere que ésta sea accesible únicamente por las entidades autorizadas.

Además en Linux se deben proteger ciertos archivos que contienen información de los usuarios (/etc/passwd, /etc/shadow), los archivos de configuración del sistema (los contenidos en etc.), el acceso al sistema y la correcta utilización de los recursos.

En muchas ocasiones los peores ataques al sistema los realiza el administrador, cuando por descuido, ignorancia o exceso de confianza altera el sistema de forma inapropiada.

Para evitar estos problemas solo hay que seguir ciertas normas de prevención como:

- No usar la cuenta de root como predeterminada, intentar primero cualquier acción como un usuario normal, si no tiene permiso puede hacer uso del comando "su" si es necesario.
- Ejecutar los comandos de forma segura verificando previamente la acción que se va a realizar.
- Activar la opción (-i) para actuar de forma interactiva con ciertos comandos que la admiten.

Al administrar un sistema Linux se deben tomar ciertas medidas que garanticen un mínimo de seguridad y funcionalidad, en primer lugar se puede limitar el espacio asignado a los usuarios ya que un posible ataque a cualquier sistema es intentar consumir todo el espacio del disco duro, una medida de seguridad es separar el árbol de directorios en diversas particiones. Pero esto puede no ser suficiente y por eso el núcleo del sistema proporciona la posibilidad de controlar el espacio de almacenamiento por grupo o usuario.

Es muy probable que muchas de las intrusiones puedan ser realizadas por los alumnos, por lo que se debe proteger al sistema de operaciones accidentales debidas a descuidos o ignorancia de parte ellos.

Una norma básica de seguridad radica en la asignación a cada usuario sólo de los permisos necesarios para poder cubrir las necesidades de su trabajo sin poner en riesgo el trabajo de los demás.

Para esto Linux dispone de todas las características de los sistemas Unix, ya que se mantiene un control de acceso a los usuarios donde se verifica login y password; por lo que se debe asegurar que los usuarios utilicen claves sólidas que cumplan con las siguientes características:

- No debe ser una palabra conocida
- Debe de contener letras, números y caracteres especiales
- Deben ser fáciles de recordar y difíciles de adivinar.

Para mantener un sistema seguro pero funcional, se tienen que otorgar los permisos adecuados para que los usuarios puedan tener acceso solo a los recursos autorizados, considerando los permisos de propietario, grupo y otros. Afortunadamente Fedora Core tiene por defecto los valores que son suficientes para mantener el sistema seguro.

Una vez que se han visto las características generales de seguridad, lo que queda es aplicar el sentido común, se tiene que ver la situación actual y responder a una serie de preguntas:

- ¿Qué se quiere proteger?
- ¿Qué valor tiene lo que se quiere proteger?
- ¿Qué costo tiene la seguridad?
- ¿De quién se quiere proteger?
- ¿Cuáles son los puntos débiles del sistema?

Después de responder estas preguntas, el administrador del sistema obtendrá un esquema elemental de la situación actual y de las medidas de seguridad básicas que debe tomar.

5.3 Trabajo futuro

Se considera que lo expuesto en esta tesis da la pauta para continuar la investigación en el área del software libre, en especial de las redes de terminales gráficas; a continuación se darán algunas ideas que pueden tomarse como base para trabajos futuros y para mejorar los resultados obtenidos:

- Se propone en primer lugar la mejora de las características del servidor para que éste pueda dar soporte a un mayor número de terminales. De acuerdo a las pruebas que se efectuaron, se pudo observar que la memoria es determinante en el rendimiento de la red, por lo que se recomienda tener 1 GB si se requiere expandir la red de 3 a 10 terminales. También es recomendable utilizar como medio de almacenamiento un disco duro SCSI en lugar de un disco IDE, ya que permite transferencias de datos con una mayor velocidad y de manera más confiable; éste aspecto es importante debido a que en un momento determinado todas las terminales pueden estar utilizando este recurso, por lo que el rendimiento de la red puede disminuir.

- Continuando con la parte de optimización de la red, se propone la implementación del proyecto PXES, que es una alternativa al LTSP, pero a diferencia de éste último permite aprovechar los recursos de las terminales con mejores capacidades, lo que

implica que se puedan correr aplicaciones localmente utilizando su memoria y disco duro, disminuyendo considerablemente la carga al servidor. Cabe mencionar que las terminales con menores capacidades como las 486, pueden seguir con el esquema del LTSP, ya que como se pudo observar en este trabajo, es el óptimo para este tipo de computadoras.

- Una posible aplicación de la red implementada en el laboratorio, es la creación de un cluster; esta arquitectura permite utilizar los recursos de las terminales para ejecutar operaciones de cálculo complejas que requieren mayor tiempo de procesamiento. El software que se propone utilizar es Open Mosix, el cual incorpora al kernel de Linux un potente algoritmo de balanceo de cargas para dividir el trabajo de procesamiento entre todos los equipos. Se debe considerar que las terminales deben poseer características de procesamiento y memoria aceptables para que el tiempo en resolver un problema específico sea mínimo.

- Otro proyecto interesante que se propone es el de la realización de una página Web que permita controlar remotamente dispositivos electrónicos conectados a las terminales, por ejemplo circuitos basados en microcontroladores. Esto permitiría entre otras cosas, monitorear el comportamiento de algunos procesos ligados a estos circuitos los cuales tienen potencialmente muchas aplicaciones. Con esto se pretende ofrecer una infraestructura que permita impulsar a los estudiantes a que desarrollen proyectos de electrónica cada vez más novedosos y funcionales.

- Finalmente, aunque es un proyecto independiente a la institución; con la experiencia adquirida en el desarrollo de esta tesis, se pretende implementar el LTSP en un cybercafe, se propone invertir en un servidor que pueda dar soporte a más de 15 computadoras 486 equipándolas con software de todo tipo, así como de Internet. La renta de los equipos de computo sería realmente mínima tomando en cuenta que pueden proporcionar los mismos resultados que si se utilizaran computadoras Pentium III, con la diferencia de brindar el mismo servicio pero a un menor precio, esto debido al ahorro de licencias, mantenimiento de los equipos, etc., y a su vez las personas se familiarizan con el uso de Linux. Posteriormente se pretende crear un servidor de música para dichas terminales, así como un servidor Samba que permita compartir recursos y archivos con diferentes plataformas de Windows (95, 98, NT, 2000, Me) y de esta manera dichas plataformas puedan acceder a los archivos Linux. El reactivar estas computadoras 486 para darle nuevas aplicaciones, y alargar su ciclo de vida; permitirá tener un negocio rentable con inversiones mínimas.

CONCLUSIONES

La idea básica del proyecto fue rescatar computadoras relativamente obsoletas como las 486 y pentium brindándoles una nueva apariencia de procesamiento gracias a la dependencia hacia un servidor, el cual les brinda sus recursos para lograr obtener unas terminales totalmente funcionales que cuentan con una buena velocidad de procesamiento, acceso a internet, procesadores de texto, software de electrónica, hasta diseños en 3D, etc., y todo bajo un ambiente gráfico totalmente amigable.

Se pudo corroborar todo esto ya que las terminales fueron probadas con éxito en el laboratorio, al principio se experimentó la facilidad de instalación del sistema operativo y una vez que se implementó el LTSP, el mantenimiento así como la actualización del software se efectuaron de una forma sencilla. El sistema realmente demostró su estabilidad y confiabilidad, nos ofrece la libertad de poder ajustarlo a nuestras necesidades, aunado a muchas otras ventajas y beneficios que se encontraron al implementar este proyecto.

También se logró recopilar una gran variedad de software de electrónica totalmente gratuito, el cual ofrece nuevas posibilidades de desarrollo para los alumnos. Además éste tipo de software permite un gran ahorro en licencias, así como de hardware adicional debido a que estas aplicaciones no necesitan de mayores recursos para funcionar adecuadamente.

Además este tipo de terminales bajo Linux no recibe tantos ataques de virus como en Windows, podemos decir que este tipo de sistema es realmente recomendable no sólo para instituciones públicas o pequeñas empresas, sino también para grandes empresas que deseen adquirir un sistema potente, seguro, y muy económico. No requiere de un gasto adicional para capacitación ya que Linux es muy intuitivo por su ambiente gráfico.

Es verdad que en Linux existen ciertos problemas con algunos modems, tarjetas de video e impresoras; donde para obtener una buena configuración se requerirá de varias horas, pero también tenemos la opción de que Linux permite ejecutar muchos programas de Windows mediante emuladores.

Con respecto al tiempo de implementación no hay mayor problema, ya que al hallar la configuración adecuada para una terminal las otras configuraciones son muy similares, esto reduce en gran medida el tiempo de implementación ya que solo es cuestión de configurar ciertas partes.

Al hacer uso de la protección de la memoria entre procesos, de manera que ninguno de ellos detenga el sistema como suele suceder usualmente en Windows, se obtiene una estabilidad determinante para el sistema así mismo cuenta con una mejor

optimización de los recursos de la computadora brindándonos la confiabilidad de realizar varios procesos sin problemas; es un sistema mucho más robusto sobre todo para aplicaciones de alta demanda.

Se optó por utilizar el proyecto LTSP debido a que nos permite aprovechar los recursos de computadoras de baja capacidad, si se contará con computadoras medianamente más potentes se pudó haber utilizado por ejemplo el proyecto de PXES ya que permite aprovechar los recursos de las terminales con mejores capacidades, lo que implica que se puedan correr aplicaciones localmente utilizando su memoria y disco duro, disminuyendo considerablemente la carga al servidor. Es por eso que promovemos el uso de un servidor más potente para lograr atender las peticiones de las terminales ya que éstas cuentan con pocos recursos, y dependen de las características del servidor.

Comprobamos que se eligió la mejor opción para reutilizar a las 486 disponibles, logrando darles nuevamente la funcionalidad que se buscaba así como nuevos campos en los que pueden ser utilizadas estas terminales bajo el concepto del LTSP.

REFERENCIAS

RAYA José Luis, RAYA Cristina, Redes Locales, Ed. Alfaomega Ra-Ma, 2002

FLORIANO Blanco Lino, GALAN Galán Susana, Al día en una hora Linux, Ed. Anaya Multimedia, 1997

CARLING M. DEGLER Stephen DENNIS James, Administración de sistemas Linux, Ed. Prentice Hall

PERRY Hekman Jessica, Linux in a nutshell. A desktop quick reference, Ed. Perry O'reilly, 1997

HUSAIN Kamran, PARKER Tim, Linux unleashed, Ed. Sams published, 1995

CARD Remy, DUMAS Erick, MEVEL Frank, The Linux kernel book, Ed. John Wiler & Suns, 1998

BLANCO J. Vicente, Linux Instalación, administración y uso del sistema, Ed. Ra-ma, 1996

NEGUS Christopher, La biblia de Red Hat Linux 7, Ed. Anaya multimedia, 2001

ESPINOSA H. Juan Carlos, Red Hat Linux 7.0 Instalación y configuración básica, Ed. Alfaomega, 2001

BENTSON Randolph, Inside Linux. A look at operating system development, Ed. Specialized System Consultants, Inc. 1996

LES Fred, DERFLER J. Jr. Frank, Guía de Novel Netware, Ed. Anaya multimedia, 1993

STAN Kelly – Bootle, Como usar Unix sistema V, versión 4.0, Ed. Noriega editores, 1993

MEDIAVILLA Manuel, Seguridad en Unix, Ed. Ra-ma, 1998

WELSH Matt, KAUFMAN Lar, Running Linux, Ed. O'reilly & associates, INC, 1996

STEFAN Strobel, VOLKER Elling, Linux unleashing the workstation in your PC, Ed. Springer, 1997

Página web de la FSF, Sección de Filosofía http://www.fsf.org/philosophy/free-sw.es.html

Página web del Proyecto GNU, Sección de Historia
http://www.gnu.org/gnu/thegnuproject.es.html

Página web del Proyecto GNU, Sección de Licencias
http://www.gnu.org/copyleft/gpl.html

Página web del Proyecto LTSP, Sección de Documentación
http://www.ltsp.org/documentation/ltsp-3.0-4-es.html

Página web de creación de imágenes de arranque http://www.rom-o-matic.net

Página web de Fedora Core http://fedora.redhat.com/download/

Página web del proyecto K12LTSP, Sección de Instalación
http://k12ltsp.org/install.html

Página web acerca de estaciones gráficas diskless
http://www.logiclinux.com/soluciones/diskless/

Página web acerca de experiencias con el LTSP http://www.cignux.org.ar/otkraus.htm

Página web de referencia para implementar un cybercafe
http://webmaster.bankhacker.com/ayuda-linux/consulta/Instalar+Cybercafe/

GLOSARIO

ASCII: (American Standard Code for Information Interchange). Es el código estándar de conjunto de caracteres que cualquier computadora puede entender, usado para representar las letras latinas, en mayúsculas, minúsculas, números, puntuación, etc. Hay 128 códigos estándar ASCII, cada uno de los cuales puede representarse por un número binario de 7 dígitos. Sin embargo, otros conjuntos de caracteres como Latin-1 están comenzando a usarse.

Broadcast: Es un tipo especial de dirección IP usada para enviar un mensaje a todos los nodos de una red.

Cliente: Programa que se usa para contactar y obtener datos de un programa de servidor localizado en otra computadora, generalmente a gran distancia. Cada programa cliente está diseñado para trabajar con uno o más tipos de programas servidores específicos, y cada servidor requiere un tipo especial de cliente.

Demonio: Es un proceso independiente encargado de llevar a cabo una tarea sin la intervención del usuario.

Dirección IP: Dirección que identifica a cada computadora en Internet utilizando una fila de cuatro conjuntos de números separados por puntos.

Diskless: PC o estación de trabajo conectado a una LAN sin disco duro ni disquetes y que depende del almacenamiento en disco de un servidor de archivos conectado a la misma LAN.

DNS: (Domain Name System). Es un sistema que traduce un nombre de dominio a un número IP, y se emplea para facilitar el manejo de direcciones ya que es más fácil recordar un nombre que un grupo de números.

EMACS: (Editing MACroS o Extensible MACro System). Es un editor muy popular para Unix y otros sistemas operativos.

Ethernet: Es la tecnología de red de área local más ampliamente utilizada. Es un estándar de la industria muy extendido, desarrollado originalmente por Xerox y formalizado en 1980 por DEC, Intel y Xerox. Las redes Ethernet transmiten datos a 10/100/1000/ Mbps utilizando un protocolo especificado.

FTP: (File Transfer Protocol). Permite transmitir archivos sobre Internet entre una máquina local y otra remota.

Host: Es una computadora anfitriona que permite a varios usuarios correr aplicaciones o programas diferentes.

Runlevel: Es el estado general de un sistema operativo tipo Unix, el cual es definido por el proceso init.

Script: Es un programa o secuencia de instrucciones que son interpretadas o ejecutadas por otro programa.

Sistema de archivos: Un sistema de archivos es el método con el que la información es introducida en un disco duro. Sistemas operativos distintos generalmente usan sistemas de archivos diferentes.

TCP/IP: (Transmisión Control Protocol/Internet Protocol) familia de protocolos que hacen posible la interconexión y trafico de red en internet.

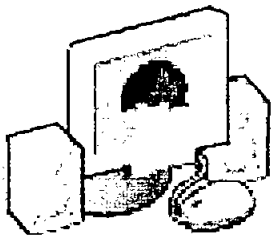
Telnet: Remote login es un protocolo que forma parte de la suit de protocolos TCP/IP y que permite conectar una terminal remota a su computadora host.

VESA: Es un estándar de modos de video para tarjetas VGA y superiores, que permite programar drivers compatibles con todas las tarjetas graficas que cumplan estas normas, independientemente del chip que incorporen.

IMPLEMENTACIÓN

OBJETIVO ESPECÍFICO

Conocer las características y consideraciones generales para la instalación y configuración de los diferentes componentes tanto de hardware como de software, que intervienen en la implementación del LTSP.



3

INTRODUCCIÓN

En este capítulo se profundizará en los conocimientos técnicos específicos, así como las herramientas que serán necesarias para entender el funcionamiento, implementación y mantenimiento del LTSP.

Es importante definir lo que se necesita para comenzar con la implementación del proyecto, una parte fundamental es la elección de la máquina servidor, ésta debe contar con un buen procesador, suficiente memoria y un disco duro de capacidad aceptable. Principalmente se prevé todo esto para un mejor rendimiento, tomando en cuenta los servicios que va a ofrecer y a cuantas terminales podrá atender eficientemente.

Se enfatiza la forma de cómo redimensionar el disco duro para no perder la información que se tiene en Windows, debido a que se permitirá que el servidor preste servicios de estación de trabajo si así se requiere. Hay que recordar que una de las características importantes de Linux es que es capaz de leer y escribir en sistemas de archivos FAT, VFAT, FAT32, además de que encontramos infinidad de aplicaciones de libre distribución que se encuentran disponibles.

Se mostrará paso a paso cómo instalar Linux especificando las opciones que son requeridas para el ambiente LTSP, así como las que mejor se adapten a las necesidades propias del laboratorio.

Finalmente se instalará y configurará el LTSP, proporcionando las condiciones adecuadas para la siguiente fase del proyecto.

3.1 Requerimientos

El LTSP es un proyecto que requiere de ciertos componentes, los cuales deben cumplir con especificaciones que permitirán su adecuado funcionamiento y rendimiento al trabajar conjuntamente, para ello es necesario definir algunos requerimientos que permitan dar una referencia de lo que realmente se necesita para su implementación. A continuación se mencionarán los requerimientos recomendados de hardware y software.

3.1.1 Hardware

Una red de terminales gráficas debe contar con un servidor que cumpla con ciertas especificaciones, para que éste pueda atender de forma eficiente las peticiones que se le hagan. Las terminales demandan recursos del servidor para poder funcionar y por lo tanto su rendimiento depende de las capacidades que el servidor pueda ofrecer.

Existen otros factores que también determinan los requerimientos físicos del servidor, como son: el tipo de programas o servicios que se van a ejecutar, el número de usuarios que van a utilizar las terminales, la arquitectura de la red, entre otros.

Tomando en cuenta estos factores, se mencionarán los requerimientos de hardware recomendados para el servidor. A pesar de que con otras configuraciones se pueden obtener buenos resultados, incluso con un servidor de menor capacidad; es importante considerar que un servidor de mejores características implica un mayor rendimiento en las terminales.

Para un servidor con 5 terminales, los requerimientos recomendados son:[12]

- CPU Intel PIII, 1 GHz
- 512 MB RAM
- Disco duro 20 GB, IDE ATA 100/133
- (2) tarjetas de red Ethernet base 10/100
- 1 CD-ROM
- 1 Floppy

Para las terminales los requerimientos recomendados son:

- CPU 486 DX, 40MHz
- 16 MB RAM
- Tarjeta de red Ethernet base 10
- Tarjeta de video VGA
- 1 Floppy

[12] Pág. web del proyecto K12LTSP, Sección de Instalación http://k12ltsp.org/install.html

Para implementar un servidor LTSP se requiere tener al menos 256 MB de RAM, más 50 MB por cada terminal; algunas aplicaciones, por ejemplo las que manipulan imágenes, necesitarán más memoria por lo que se recomienda tener 1 GB.

Si la red tiene más de seis terminales, es recomendable contar con discos SCSI; estos discos se caracterizan por que son mucho más rápidos que los IDE, además de que permiten accesos múltiples a los mismos.

La velocidad del procesador también depende del uso que se le dará a los equipos; por ejemplo las aplicaciones de ofimática requieren de una velocidad relativamente pequeña (1 GHz). Al igual que con la memoria, las aplicaciones gráficas necesitan una mayor velocidad del procesador.

En cuanto a las tarjetas de red, las de 100 Mb funcionan bastante bien para una red que tenga hasta 30 terminales; además, es necesario contar con un Switch o un Hub, que permita implementar el modelo cliente-servidor. Se recomienda usar un Switch, ya que tiene control sobre colisiones y esto disminuye el tráfico en la red.

En el laboratorio de electrónica hay computadoras que satisfacen algunas de estas características; por lo que no será necesario adquirir algún equipo nuevo, además la idea de implementar este proyecto es precisamente el de seguir utilizando las computadoras con las que ya se cuenta, aunque no se descarta la posibilidad de que en un futuro se puedan mejorar las características del servidor.

Las características de este equipo son las siguientes:

Equipo 1

- CPU Intel 486, 66 MHz
- 16 MB RAM
- Disco duro 512 MB, IDE ATA 100
- Tarjeta de red 3Com 509 EtherLink III, bus ISA

Equipo 2

- CPU Intel 486, 66 MHz
- 16 MB RAM
- Disco duro 512 MB, IDE ATA 100
- Tarjeta de red 3Com 509 EtherLink III, bus ISA

Equipo 3

- CPU Intel PIII, 700 MHz
- 128 MB RAM
- Disco duro 20 GB IDE ATA 100
- Tarjeta de red MX98715, bus PCI

Equipo 4

- CPU Intel PIII, 700 MHz
- 384 MB RAM
- Disco duro 20 GB IDE ATA 100
- Tarjeta de red Sis 900, bus PCI
- Tarjeta de red RTL8139C, bus PCI

Se optó por utilizar el último equipo como servidor debido que cuenta con las mejores características.

3.1.2 Software

La computadora que será utilizada como servidor ya tiene instalado el sistema operativo Windows, sin embargo éste no reúne las características para implementar el LTSP; por esta razón es necesario instalar otro sistema operativo, para este caso Linux ya que cuenta con capacidades tanto de estación de trabajo como de servidor.

3.1.2.1 Redimensionar la partición

Debido a que en el laboratorio se trabaja con aplicaciones que solo pueden correr bajo Windows se optó por conservarlo, éste ocupa toda la capacidad del disco duro, es decir; solo existe una partición tipo FAT 32 sin embargo Linux requiere sus propias particiones.

Para liberar espacio en el disco duro donde estarán las particiones de Linux, se requiere modificar el tamaño de la partición de Windows, para lo cual se hará uso de la herramienta llamada Partition Magic que se ejecuta desde MS-DOS o Windows y permite crear, borrar o redimensionar las particiones de un disco de manera sencilla y confiable.

A continuación se muestra el proceso para liberar el espacio en disco para Linux:

Después de ejecutar el Partition Magic se muestra la partición de Windows que ocupa los 20 GB totales, pero se necesita redimensionarla para que solo sea de 10 GB, el espacio restante será asignado a las particiones de Linux.

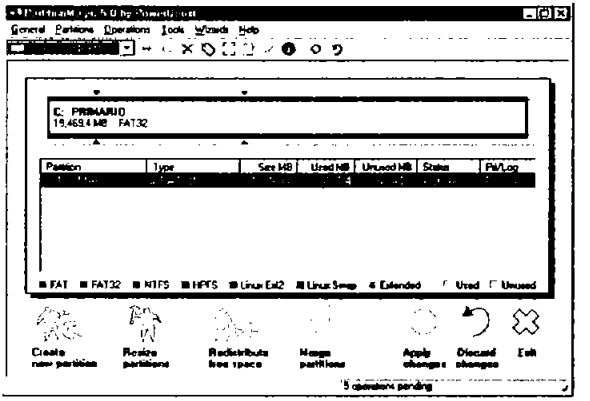


Figura 3-1 Se muestra la partición de Windows que ocupa 20 GB

Posteriormente se elige resize/move del menú operations, que muestra una ventana donde se pueden modificar las propiedades de la partición.

El tamaño de la partición se puede cambiar de 2 maneras, la primera posicionándose al final de la barra y arrastrando el puntero del ratón hasta fijar el nuevo tamaño y la segunda introduciendo el valor correspondiente en MB en el campo New Size, por último se da click en el botón OK para aceptar los cambios.

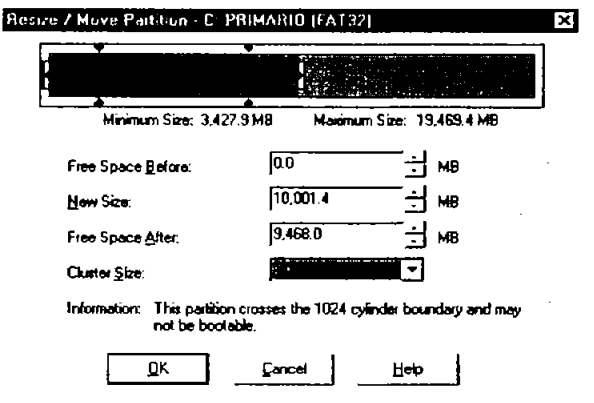


Figura 3-2 Proceso de liberación de espacio en disco para Linux

Ahora se puede observar el nuevo tamaño de la partición de Windows y el espacio libre para Linux.

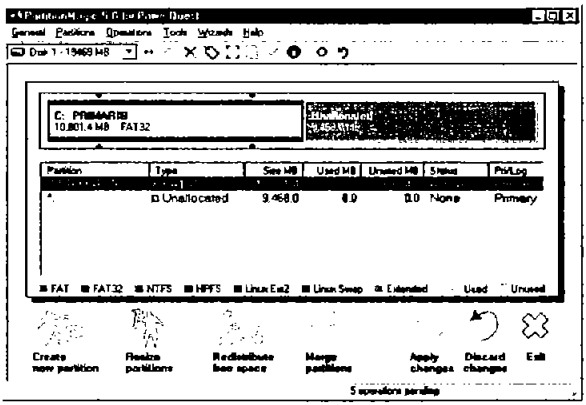


Figura 3-3 Nuevo tamaño de la partición de Windows y el espacio para Linux

Finalmente se da click en Apply Changes para efectuar los cambios, el programa tendrá que cerrar Windows para comenzar con el proceso de redimensionamiento. Una vez que haya finalizado este proceso, se podrá comenzar con la instalación de Linux. Si no se desea conservar la instalación de Windows, se puede formatear el disco duro, para que Linux ocupe todo el espacio disponible.

3.1.2.2 Instalación del sistema operativo

En esta sección se describen las opciones y configuraciones recomendadas durante la instalación de Fedora Core. En este caso el método de instalación será vía CD-ROM, por lo que es necesario configurar el BIOS de la computadora para que permita bootear desde este dispositivo.

Paso 1: Se introduce el CD 1 y automáticamente se inicia el programa de instalación de Fedora Core llamado Anaconda, en caso de que el programa no pueda ser iniciado se tiene que crear un disco de arranque, éste permitirá cargar una imagen del sistema operativo con la información necesaria para poder realizar la instalación por medio del CD-ROM.

El disco de arranque se crea de la siguiente manera:

Desde MS-DOS, se utiliza el comando rawrite, el cual viene incluido en el primer disco de instalación de Fedora Core, en la carpeta DOSUTILS.

Se escribe el comando:

```
D:\DOSUTILS\rawrite
```

Se pedirá la ruta de la imagen de arranque:

```
D:\IMAGES\boot.img
```

Y después la ruta donde se escribirá la imagen de arranque, en este caso es el disquete:

```
A:\
```

Se tiene que reiniciar la computadora para que ésta arranque por medio del disco generado.

Paso 2: El programa de instalación muestra una pantalla de inicio, donde no se pide información alguna. La sección de ayuda aparece por defecto, es recomendable leer la información mostrada.

Para continuar con la instalación, se debe hacer click en el botón "Siguiente".

Paso 3: Seleccionar el idioma español, éste es el lenguaje que será utilizado para la instalación y para el sistema. Hacer click en "Siguiente" para continuar.

Paso 4: Seleccionar el tipo de teclado apropiado para el sistema, en este caso se elige "español". Hacer click en "Siguiente" para continuar.

Paso 5: Se debe seleccionar el tipo de ratón, en la mayoría de los casos será detectado automáticamente. Si el ratón es de 2 botones, se puede habilitar la opción para emular 3 botones. Hacer click en "Siguiente" para continuar con el proceso.

Paso 6: Debido a la flexibilidad que se requiere para la instalación de un servidor de terminales gráficas, se debe seleccionar el tipo de instalación más apropiado; en este caso será la instalación "personalizada". Para continuar, se debe hacer click en "Siguiente".

Paso 7: En este punto se tiene que elegir una de las opciones de particionamiento del disco duro. Seleccionar "Disk Druid" y hacer click en "Siguiente".

Paso 8: Se debe tener cuidado al crear las particiones ya que de esto depende el buen funcionamiento del sistema.

Disk Druid muestra gráficamente las particiones del disco duro, en este caso la partición de Windows y el espacio libre destinado para Linux.

La primer partición que hay que crear es la swap, para esto se debe seleccionar con el ratón el espacio libre del disco duro, y hacer click en el botón "nuevo" lo cual abre un cuadro de diálogo donde se puede configurar las características de la nueva partición.

Se debe seleccionar en el campo "Tipo de sistema de archivos" la opción "Swap", posteriormente se especifica su tamaño, el cual será de 768 MB. Finalmente se habilita la opción "tamaño fijo". Hacer click en Aceptar para cerrar el cuadro de diálogo.

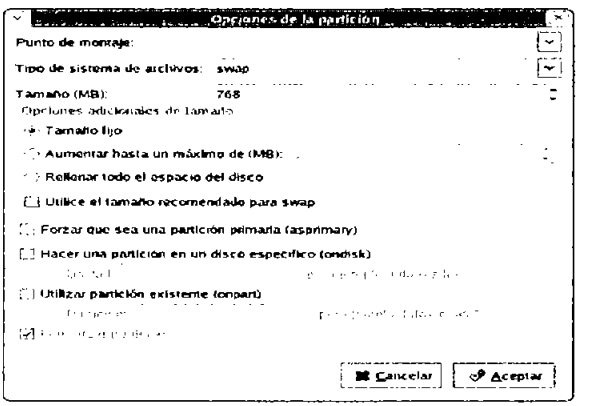


Figura 3-4 Proceso de creación de la partición swap

La segunda partición que debe ser creada es la /boot, que se utiliza para almacenar los archivos de arranque del sistema.

Con ayuda del ratón se debe seleccionar el espacio libre del disco duro, y hacer click en el botón "nuevo" lo cual abre el cuadro de diálogo para configurar la nueva partición.

En la sección punto de montaje se debe elegir "/boot" y como sistema de archivos el tipo "ext3". Esta partición tendrá un tamaño de 100 MB, por lo que se debe habilitar la opción "tamaño fijo".

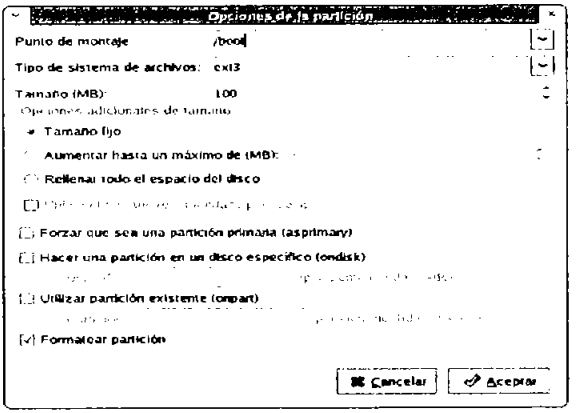


Figura 3-5 Proceso de creación de la partición boot

La última partición que se tiene que crear es la raíz (/), para esto debe seleccionarse el espacio libre del disco duro, posteriormente se hace click en el botón "nuevo" para abrir el cuadro de diálogo correspondiente.

En la sección "punto de montaje" se elige "/" y como sistema de archivos el tipo "ext3", en la sección "opciones de tamaño" se debe seleccionar "completar hasta el tamaño máximo permitido"; lo cual permite seleccionar todo el espacio libre del disco duro.

Hacer click en Aceptar para cerrar el cuadro de diálogo.

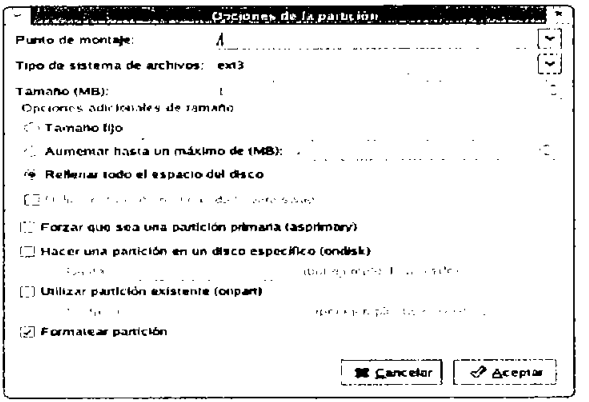


Figura 3-6 Proceso de creación de la partición raíz

Para finalizar el proceso de creación de las particiones de debe hacer click en "siguiente".

Paso 9: El programa Anaconda, instala por defecto el gestor de arranque GRUB, el cual permitirá elegir con que sistema operativo iniciar (Windows o Fedora Core). Hacer click en "Siguiente" para continuar.

Paso 10: Fedora Core permite proteger el sistema mediante un firewall, para instalar un nivel aceptable de seguridad, se debe elegir la opción "nivel intermedio". Hacer click en "Siguiente" para continuar.

Paso 11: Es necesario instalar un idioma para usarlo por defecto, éste será utilizado por el sistema una vez que la instalación se haya completado. En este caso se debe seleccionar "Español (México)". Hacer click en "Siguiente" para continuar.

Paso 12: El programa de instalación permite seleccionar un huso horario, el cual será "América / Ciudad de México". Hacer click en "Siguiente".

Paso 13: En este paso se tiene que proporcionar la cuenta root, la cual es usada para instalar paquetes, actualizar RPM's y realizar la mayoría de las tareas de mantenimiento del sistema. Una vez que se haya ingresado y confirmado la contraseña, se debe hacer click en "Siguiente".

Paso 14: En esta parte es donde se eligen los paquetes que serán instalados en el sistema.

A continuación se muestra una lista con los grupos de paquetes a instalar.

- Escritorios
  - Sistema X Window
  - GNOME
  - KDE

- Aplicaciones
  - Editores
  - Ingeniería y científico
  - Internet gráfica
  - Oficina/productividad
  - Sonido y video
  - Gráficos

- Servidores
  - Servidores de red (elegir DHCP)

- Desarrollo
  - Herramientas de desarrollo
  - Desarrollo del kernel

- Desarrollo del software X
- Desarrollo de software de GNOME
- Desarrollo de software para KDE

➢ Sistema
- Herramientas de administración
- Soporte para la impresión

Paso 15: El programa de instalación formateará las particiones e iniciará la instalación de los paquetes necesarios. Durante esta etapa, se le solicitará el ingreso de los CD's 2 y 3.

Paso 16: Una vez finalizado el proceso de instalación de los paquetes, se puede crear un disco de arranque; este disco se utiliza en caso de que el gestor de arranque GRUP no funcione. Hacer click en continuar.

Paso 17: Una vez que se hayan completado todos los pasos anteriores, el programa finalizará la instalación.

3.2 Instalación y configuración de los paquetes en el servidor.

Hasta ahora se conoce de manera general cual es la forma en que trabaja el LTSP, sin embargo es necesario comprender a fondo todo lo que implica la implementación de este proyecto; para ello se describirá la función y la instalación de cada uno de los programas que deben estar presentes en el servidor previamente a los paquetes del LTSP. Posteriormente se mostrará la forma en que deben ser configurados para que funcionen en conjunto con el LTSP.

Se da por hecho que el servidor ya debe estar corriendo el sistema operativo Fedora Core, y que se cuenta con los permisos de root.

3.2.1 DHCP

Dynamic Host Configuration Protocol (Protocolo de Configuración Dinámica de Servidor) es un protocolo de red que se utiliza para asignar automáticamente información TCP/IP a equipos cliente. Cada uno de ellos se conecta a un servidor DHCP centralizado que devuelve la configuración de red del cliente, incluida la dirección IP, el gateway y los servidores DNS.

DHCP es útil para proporcionar de un modo rápido la configuración de red de una máquina cliente, al configurar el sistema cliente el administrador puede seleccionar

el protocolo DHCP y no especificar una dirección IP, una máscara de red, un gateway o servidor DNS fijos; el cliente recupera esta información desde el servidor.

DHCP también es útil si un administrador desea cambiar la dirección de IP de muchos sistemas; en lugar de volver a configurarlos todos, puede modificar un archivo de configuración en el servidor para establecer la nueva dirección IP.

Si los servidores DNS de una organización cambian, los cambios también se aplicarán en el servidor DHCP, no en todos los clientes. Una vez que se reinicie la red, se aplicarán los cambios.

Además, si una computadora portátil o cualquier tipo de equipo móvil se configura para hacer peticiones DHCP, podrá desplazarse entre distintas oficinas sin tener que volver a configurarse, ya que cada oficina dispondrá de un servidor DHCP que permitirá su conexión a la red.

3.2.1.1 Instalación de los paquetes RPM

Para la instalación del servicio DHCP, se debe descargar la última versión de los paquetes y ejecutar los siguientes comandos:

```
rpm -ivh dhcp-devel-3.0pl1-9.i386.rpm
rpm -ivh dhcp-3.0pl1-9.i386.rpm
```

3.2.2 TFTP

Trivial File Transfer Protocol (Protocolo Simple de Transferencia de Archivos) es un protocolo que utiliza el UDP (Protocolo de Datagramas de Usuario) para transferir archivos entre distintas máquinas conectadas a través de una red.

Se implementa sobre un servicio de comunicaciones no fiable y no orientado a conexión, por lo que no requiere el nombre de un usuario ni su contraseña, lo que da lugar a que carezca de seguridad. TFTP ejecuta las mismas tareas que el FTP (Protocolo de Transferencia de Archivos), pero usa un protocolo de transporte diferente.

3.2.2.1 Instalación de los paquetes RPM

Para la instalación del servicio TFTP, se debe descargar la última versión de los paquetes y ejecutar los siguientes comandos:

```
rpm -ivh tftp-0.29-3.i386.rpm
rpm -ivh tftp-server-0.29-3.i386.rpm
```

3.2.3 NFS

Network File System (Sistema de Archivos de Red) es un método para compartir archivos entre máquinas de una red, de tal forma que parece que se está trabajando en un disco duro local. Esto permite centralizar archivos en una localización, mientras se permite su acceso continuo a los usuarios autorizados.

Fedora Core puede trabajar como servidor o como cliente de NFS (o ambos), lo que implica que puede exportar sistemas de archivos a otras máquinas, así como montar los sistemas de archivos que otras computadoras exportan.

NFS resulta útil para compartir directorios de archivos entre múltiples usuarios de la misma red. Por ejemplo, un grupo de usuarios que trabajan en un mismo proyecto pueden tener acceso a los archivos del mismo usando una porción compartida del sistema de archivos NFS, que se ha montado en un directorio determinado, como pudiera ser /myproject. Para acceder a los archivos compartidos; el usuario accede al directorio /myproject de su máquina local.

Linux usa una combinación de soporte a nivel de kernel y demonios en continua ejecución para proporcionar la disponibilidad de los archivos vía NFS, y el servicio debe estar activo en el kernel de Linux para que funcione. NFS usa Remote Procedure Calls (Llamadas de Procedimientos Remotos) para enrutar peticiones entre clientes y servidores, implicando que el servicio portmap deba estar disponible y activo en los niveles de ejecución adecuados para que la comunicación funcione.

Con NFS, la autentificación solo se produce cuando el cliente intenta montar un sistema de archivos remoto. Para limitar el acceso, el servidor NFS utiliza los archivos /etc/hosts.allow y /etc/hosts.deny para determinar si a un cliente en particular le debe ser explícitamente permitido o denegado su acceso.

Después de revisar los archivos /etc/hosts.allow y /etc/hosts.deny, el servidor NFS recurre a su archivo de configuración, /etc/exports, para determinar si el cliente tiene suficientes privilegios para montar alguno de los sistemas de archivos exportados. Los privilegios de montaje NFS son permitidos específicamente a clientes, no a usuarios. Si se permite a una máquina acceder a un sistema de archivos exportado, los usuarios de esa máquina podrán acceder a esos datos.

3.2.3.1 Instalación de los paquetes RPM

Para la instalación del servicio NFS, se debe descargar la última versión de los paquetes y ejecutar los siguientes comandos:

```
rpm -ivh nfs-utils-1.0.1-2.i386.rpm
rpm -ivh portmap-4.0-46.i386.rpm
```

3.2.4 LTSP

Existen una serie de paquetes para instalar el entorno LTSP, hay 3 grupos principales los cuales son: Core, Linux Kernel y X Windows. Se dividieron los paquetes en grupos, para mejorar su funcionalidad y flexibilidad.

3.2.4.1 Paquete Core

Este paquete proporciona los scripts de instalación del LTSP, los cuales permiten crear la estructura de directorios que será compartida a cada una de las terminales y también los archivos plantilla para la configuración de los servicios requeridos.

Se deberá conseguir la última versión del paquete y ejecutar el siguiente comando:

```
rpm –ivh ltsp_core-3.0.7-4.i386.rpm
```

3.2.4.2 Paquete Ltsp_Kernel

Este paquete contiene un kernel con un ramdisk (disco virtual). Dentro del ramdisk se incluye soporte para todos los módulos de las tarjetas de red y un espacio de usuario con un cliente dhcp llamado dhclient.

Se deberá conseguir la última versión del paquete y ejecutar el siguiente comando:

```
rpm –ivh ltsp_kernel-3.0-5.i386.rpm
```

3.2.4.3 Paquete X Windows

Este paquete es necesario si se desea configurar una terminal gráfica (Interfaz gráfica GUI), el cual proporciona todos los drivers pertenecientes al XFree86; es recomendable que también se instale el paquete de fuentes X Fonts. Este paquete viene separado del X Core debido a que en la mayoría de los casos se utiliza un servidor de fuentes X (Xfs).

Se deberá conseguir la última versión de los paquetes y ejecutar los siguientes comandos:

```
rpm –ivh ltsp_x_core-3.0.4-2.i386.rpm
rpm –ivh ltsp_x_fonts-3.0.4-2.i386.rpm
```

3.2.4.4 Scripts de configuración LTSP

Después de instalar los paquetes principales se requiere ejecutar algunos scripts que son necesarios para configurar los archivos de sistema del servidor. Cada uno de estos scripts es responsable de modificar o en su caso crear archivos de ejemplo de los servicios NFS, TFTP, DHCP, etc.

Se accede al directorio install_scripts con el siguiente comando:

```
cd /opt/ltsp/install_scripts
```

Posteriormente se ejecuta el siguiente script para crear el directorio templates.

```
sh ./install.sh
```

Este script muestra la siguiente salida:

```
About to install LTSP, using the following settings:

 LTSP_DIR    = /opt/ltsp
 SWAP_DIR    = /var/opt/ltsp/swapfiles
 TFTP_DIR    = /tftpboot
 IP_NETWORK  = 192.168.0.0
 IP_SERVER   = 192.168.0.254
 IP_NETMASK  = 255.255.255.0
 IP_BROADCAST = 192.168.0.255

If you want to install LTSP using the above settings,
enter 'Y' and the installation will proceed.  Any other
response will abort the installation, and you can modify
the CONFIG file and restart the installation.

Continue with installation (y/n)?

find: i386: No existe el archivo o el directorio
./install.sh: line 274: /opt/ltsp/i386/etc/version: No existe el archivo o el directorio

Take a look in /tmp/ltsp.install.log for a complete log of the installation

You now need to change to the /opt/ltsp/templates directory and
run the ltsp_initialize script to complete the installation

./install.sh: line 397: /opt/ltsp/i386/etc/lts.orig: No existe el archivo o el directorio
```

Tabla 3-1 Creación del directorio templates

Aunque se muestran ciertos errores, estos pueden ignorarse; lo importante en este punto es que se haya creado el directorio templates.

Una vez creado el directorio se tendrá que entrar a él.

```
cd ../templates
```

Estando en el directorio templates se ejecuta lo siguiente:

```
./ltsp_initialize
```

Este comando preguntará qué servicios se quieren configurar, para ello tendrá que crear o modificar ciertos archivos. La salida que muestra este comando es la siguiente:

```
Linux Terminal Server Project (http://www.LTSP.org)

The following files will be created/modified:

/etc/X11/xdm/Xaccess              The config file to allow remote xdm log  [Y]
/etc/X11/xdm/Xsetup_workstation   Sets the logo of your login window       [Y]
/etc/dhcpd.conf.example           Example config file for dhcp             [Y]
/etc/exports                      The config file for nfs                  [Y]
/etc/X11/gdm/gdm.conf             The config file for gdm                  [Y]
/etc/X11/gdm/Init/Default         The gdm startup script                   [Y]
/etc/hosts.allow                  Configuration file for tcp wrappers      [Y]
/etc/inittab                      Config file for init                     [Y]
/etc/kde/kdm/kdmrc                The config file for kdm                  [Y]
/etc/X11/xdm/ltsp.gif             The background logo for your login       [Y]
/etc/rc.d/rc5.d/S60nfs            Startup links for nfs                    [Y]
/etc/rc.d/rc5.d/S13portmap        Startup links for portmapper             [Y]
/etc/sysconfig/syslog             Startup info for syslogd                 [Y]
/etc/xinetd.d/tftp                Enable the tftp daemon                   [Y]
/etc/X11/xdm/xdm-config           The main config file for xdm/kdm         [Y]


Ready to apply the changes? ( R-Review, A-Apply, C-Cancel )
```

Tabla 3-2 Ejecución de scrip ltsp.initialize en el directorio templates

Debido a que en ocasiones algunos de estos archivos son configurados manualmente por el administrador por cuestiones de seguridad, es recomendable verificar las modificaciones que realiza este script. En caso de estar convencidos de que no se va a comprometer la seguridad del sistema, se pueden aplicar los cambios; lo que generará o en su caso modificará los archivos mencionados.

3.2.5 Configuración de los archivos de sistema

Los scripts del LTSP, crearán entradas (líneas de código) en los archivos de configuración de los diferentes servicios necesarios; estas entradas son configuraciones por default de lo que se necesita para que el servidor LTSP funcione adecuadamente. Algunos parámetros podrán modificarse o adaptarse dependiendo de la configuración de la red, como por ejemplo las direcciones IP; sin embargo es recomendable conservar los valores iniciales. En algunos casos se crearán líneas comentadas, las cuales servirán de ejemplo en caso de requerir configuraciones especiales, esto sucede a menudo debido a la gran diversidad del hardware disponible. A continuación se muestran los archivos de sistema más importantes para la implementación del LTSP.

3.2.5.1 dhcpd.conf

Este archivo de configuración consta de un conjunto de sentencias, las cuales se clasifican en parámetros y declaraciones. Los parámetros expresan como hacer algo, si se hace algo o no, así como los atributos que se le asignan al cliente. Las declaraciones, en cambio, se emplean para describir la topología de una red, describir a un conjunto de clientes o para aplicar determinados parámetros a un grupo de declaraciones.

Las declaraciones tienen la forma:

```
<nombre de la declaración> [atributos] {
        [parámetros]
        [declaraciones]
}
```

y los parámetros:

```
[option] <nombre del parámetro> [valores];
```

Los parámetros que comienzan con la palabra reservada "option" describen aquellos datos que brinda el servidor al cliente como parte del protocolo, y los que no, describen las características del servidor de DHCP.

A continuación se describen las sentencias declarativas:

- shared-network: Permite agrupar un conjunto de subredes que compartan la misma red física. El único atributo de esta sentencia es un nombre que sólo se utiliza para las trazas del servicio.

Sintaxis:

```
shared-network <nombre> {
        [parámetros]
        [declaraciones]
}
```

- subnet: Permite agrupar las características globales que van a tener los clientes de una misma subred.

Sintaxis:

```
subnet <dirección de red>netmask<máscara de red> {
        [parámetros]
        [declaraciones]
}
```

- range: Permite definir un rango de direcciones IP a otorgar a clientes pertenecientes a una subred. Toda declaración tipo subnet debe tener asociada una declaración range en la cual se especifique las direcciones IP mínima y máxima. Si se especifica el atributo dynamic-bootp se indica que estas direcciones se pueden asignar también a clientes BOOTP. Cuando se especifica una sola dirección IP se omite la dirección máxima.

Sintaxis:

```
range [dynamic-bootp] <dirección IP mínima> [dirección IP máxima]
```

- host: Permite describir aquellos hosts que tengan una dirección fija. Todos los clientes que usan BOOTP deben tener asociada una sentencia host. Un cliente se corresponde con una declaración host si la opción dhcp-client-identifier indicada en la declaración posee el valor del identificador que brinda el cliente a través del protocolo. De no ser así entonces se emplearía la dirección MAC del cliente especificada a través del atributo hardware.

Sintaxis:

```
host <hostname> {
        [parámetros]
        [declaraciones]
}
```

- group: Permite agrupar a otras declaraciones para aplicarles varios parámetros comunes. Puede ser utilizada para agrupar hosts, subredes, redes compartidas y otros grupos.

Sintaxis:

```
group {
        [parámetros]
        [declaraciones]
}
```

Los principales parámetros son:

- lease-file-name <filename>: Indica el nombre del archivo donde se almacenan los contratos. Este parámetro tiene alcance global por lo que se debe especificar fuera de todos los ámbitos (declaraciones) para que tenga efecto real. Por defecto es /var/lib/dhcp/dhcpd.leases

- default-lease-time <time>: Expresa en segundos la duración de una conexión.

- max-lease-time <time>: Expresa en segundos la duración máxima de una conexión.

- min-lease-time <time>: Expresa en segundos la duración mínima de una conexión.

- min-seconds <seconds> : Indica el número de segundos que debe esperar el servidor DHCP para responder a la petición de los clientes. Se utiliza cuando se tiene un segundo servidor y se desea que este responda después que el otro haya atendido la petición del cliente.

- hardware <type> <address>: Indica la dirección física (MAC) de un cliente particular (declaraciones tipo host). El atributo type expresa el tipo de arquitectura de la interfaz de red, actualmente puede ser: ethernet o token-ring.

- server-name <servername>: Indica el nombre que se ofrecerá a los clientes como identificador del servidor que emplean.

- fixed-address <address> [, <address>]: Expresa las direcciones IP que son fijas para los clientes descritos a través de las declaraciones de tipo host. Pueden utilizarse nombres de dominio en lugar de números IP.

- dynamic-bootp-lease-cutoff <date>: Indica la fecha en que expiran los contratos de todos los clientes BOOTP.

- get-lease-hostnames <flag>: Indica si el servidor resolverá o no las direcciones IP de los clientes a nombres de dominio y usará estos nombres como la opción host-name del protocolo.

- use-host-decl-names <flag>: Indica si se asume que el nombre provisto en cada una de las declaraciones tipo host dentro del mismo ámbito, es el nombre del cliente correspondiente (opción host-name del protocolo).

- authoritative;
- non authoritative;

Indican si el servidor está autorizado o no para realizar sus funciones. Por defecto un servidor DHCP asume que la información que brinda a una subred determinada no es correcta ni tampoco está autorizado para brindar el servicio.

Esto permite que si un usuario inexperto instala un servidor de DHCP en la red este no sea escuchado por los clientes como lo es un servidor legítimo que se le indique explícitamente que está autorizado.

El administrador de red que configure adecuadamente su servidor debe colocar este parámetro al comienzo del archivo, aunque puede ser conveniente en algunas ocasiones declarar al servidor autorizado de acuerdo a los segmentos de red definidos y no de forma global.

- always-broadcast <flag>: Se emplea para algunos clientes de DHCP/BOOTP que no pueden recibir las respuestas del servidor si no son en forma de broadcast. Se debe tratar de colocar este parámetro a "on" sólo para los clientes que realmente lo necesiten pues provoca demasiado tráfico en la red.

- ddns-update <flag>: Indica si se realizan o no actualizaciones dinámicas al DNS siempre que se establezca un contrato. Por defecto este parámetro tiene valor "on".

- allow <request>;
- deny <request>;
- ignore <request>;

Se emplean para controlar la respuesta del servidor DHCP ante distintos tipos de peticiones, algunas de las posibles solicitudes (atributo request) son:

  - unknown-clients: se emplea para indicar al servidor si acepta o no las solicitudes de los clientes desconocidos. Un cliente desconocido es aquel que no tiene asociado una declaración tipo host. Por defecto las solicitudes de estos clientes se aceptan.

- bootp: se utiliza para señalar si se aceptarán o no los pedidos de los clientes BOOTP. Por defecto se aceptan.

- booting: se emplea en las declaraciones del tipo host para indicar si se aceptará o negará la solicitud del host correspondiente. Por defecto se aceptan para todos los hosts.

- declines: se utiliza para indicar si el servidor acepta o no los mensajes del tipo DHCPDECLINE de los clientes. Cuando un servidor recibe este tipo de mensajes asume que la dirección que ofrece no es válida pues al parecer alguien no autorizado la está utilizando y entonces la declara como abandonada.

  Desafortunadamente un cliente "malicioso" o con una implementación incorrecta puede agotar todo el spool de direcciones a otorgar que posee el servidor y antes de que este decida emplear las direcciones abandonadas ya se habrán provocado algunos trastornos en el servicio.

Entre los parámetros que se le pueden otorgar a un cliente a través del protocolo y que van precedidos por la palabra "option", se encuentran:

- option domain-name <domain name>: Indica el nombre del dominio que empleará el cliente.

- option domain-name-servers <ip address> [, <ip address> ...]: Indica los servidores de nombres de dominio a emplear por el cliente.

- option host-name <hostname>: Indica el nombre que empleará el host cliente.

- option subnet-mask <ip address>: Indica la máscara de red que se le asignará al cliente.

- option routers <ip address> [, <ip address> ...]: Indica las direcciones IP de los routers (gateway) que empleará el cliente.

- option broadcast-address <ip address>: Indica la dirección de broadcast que utilizará el cliente.

- option dhcp-client-identifier <string>: Indica el identificador que puede emplear el cliente como alternativa a su dirección MAC.

Un ejemplo del archivo dhcpd.conf es creado como parte de la instalación del LTSP. Este archivo es llamado /etc/dhcpd.conf.example y puede ser copiado o renombrado a /etc/dhcpd.conf.

El archivo tiene la siguiente estructura:

```
# Sample configuration file for dhcpd
#
# Make changes to this file and copy it to /etc/dhcpd.conf
#
ddns-update-style       none;

default-lease-time          21600;
max-lease-time              21600;

option subnet-mask          255.255.255.0;
option broadcast-address    192.168.0.255;
option routers              192.168.0.254;
option domain-name-servers  192.168.0.254;
option domain-name          "yourdomain.com";
option root-path            "192.168.0.254:/opt/ltsp/i386";

option option-128 code 128 = string;
option option-129 code 129 = text;

shared-network WORKSTATIONS {
  subnet 192.168.0.0 netmask 255.255.255.0 {
  }
}

group   {
  use-host-decl-names     on;
  option log-servers      192.168.0.254;

  host ws001 {
    hardware ethernet   00:E0:06:E8:00:84;
    fixed-address       192.168.0.1;
    filename            "/lts/vmlinuz-2.4.19-ltsp-1";
##      option option-128   e4:45:74:68:00:00;  #This is NOT a MAC address
##      option option-129   "NIC=ne IO=0x300";
  }
  host ws002 {
    hardware ethernet   00:D0:09:30:6A:1C;
    fixed-address       192.168.0.2;
    filename            "/lts/vmlinuz-2.4.9-ltsp-6";
  }
}
```

Tabla 3-3 Ejemplo de configuración del archivo dhcpd

Una vez renombrado el archivo y hechas las modificaciones pertinentes el dhcpd puede ser iniciado con el siguiente comando:

```
service dhcpd start
```

3.2.5.1.1 Terminal 1

Para este equipo, la configuración del archivo dhcpd.conf debe ser:

```
host ws001 {
        hardware ethernet    00:60:8C:CA:BC:FF;
        fixed-address        192.168.0.1;
        filename             "/lts/vmlinuz-2.4.19-ltsp-1";
        option option-128    e4:45:74:68:00:00;
        option option-129    "NIC=3c509";
}
```

3.2.5.1.2 Terminal 2

Para este equipo, la configuración del archivo dhcpd.conf debe ser:

```
host ws002 {
        hardware ethernet    00:A0:24:24:F6:0F;
        fixed-address        192.168.0.2;
        filename             "/lts/vmlinuz-2.4.19-ltsp-1";
        option option-128    e4:45:74:68:00:00;
        option option-129    "NIC=3c509";
}
```

3.2.5.1.3 Terminal 3

Para este equipo, la configuración del archivo dhcpd.conf debe ser:

```
host ws003 {
        hardware ethernet    00:80:AD:3B:8F:E4;
        fixed-address        192.168.0.3;
        filename             "/lts/vmlinuz-2.4.19-ltsp-1";
}
```

3.2.5.2 tftp

El archivo tiene la siguiente estructura:

```
# default: off
# description: The tftp server serves files using the trivial file transfer\
#       protocol.  The tftp protocol is often used to boot diskless \
#       workstations, download configuration files to network-aware printers, \
#       and to start the installation process for some operating systems.
service tftp
{
        socket_type             = dgram
        protocol                = udp
        wait                    = yes
        user                    = root
        server                  = /usr/sbin/in.tftpd
        server_args             = -s /tftpboot
        disable                 = yes
        per_source              = 11
        cps                     = 100 2
        flags                   = IPv4
}
```

Tabla 3-4 Ejemplo de configuración del archivo tftp

En Fedora Core, el servicio tftp es iniciado por xinetd; el archivo de configuración tftp contiene la información necesaria para que corra tftpd. Por lo que el servicio podrá ser iniciado con el siguiente comando:

```
service xinetd start    ó
service xinetd restart
```

3.2.5.3 hosts

Por lo general las computadoras se comunican con direcciones IP, pero para las personas es más sencillo recordar nombres que números, es entonces cuando el DNS o el archivo /etc/hosts entra escena. Este mapeo de direcciones IP a nombres de host no es requerido generalmente, pero en un ambiente LTSP si lo es. Esto es así porque sin este mapeo, el NFS dará errores de permisos cuando la terminal intente montar el sistema de archivos raíz. Aparte de los problemas de NFS, si la terminal no se encuentra en el archivo /etc/hosts, también se podrían tener problemas con los Display Managers GDM y KDM.

El principal propósito de este archivo es resolver los nombres de hosts a partir de su dirección IP. Se puede usar solamente para resolver nombres de hosts en pequeñas redes sin servidor DNS.

Sin tener en cuenta el tipo de red que la computadora utilice, este archivo contiene una línea que especifica la dirección IP del dispositivo loopback (127.0.0.1) como por ejemplo localhost.localdomain; éste dispositivo hace referencia a la misma computadora.

La sintaxis de este archivo es muy simple:

```
<dirección IP> minombre.midominio.org alias
```

Se deben utilizar direcciones IP que pertenezcan al rango de direcciones IP privadas. Los rangos válidos son:

Clases de redes

A 10.0.0.0
B Entre 172.16.0.0 y 172.31.0.0
C Entre 192.168.0.0 y 192.168.255.0

El archivo tiene la siguiente estructura:

```
# Do not remove the following line, or various programs
# that require network functionality will fail.
127.0.0.1       localhost.localdomain       localhost
192.168.0.254   localhost.localdomain       localhost
```

Tabla 3-5 Ejemplo de configuración del archivo hosts

3.2.5.3.1 Terminal 1

Para este equipo, la configuración del archivo hosts debe ser:

```
192.168.0.1     ws001     ws001.localdomain     ws001
```

3.2.5.3.2 Terminal 2

Para este equipo, la configuración del archivo hosts debe ser:

```
192.168.0.2     ws002     ws002.localdomain     ws002
```

3.2.5.3.3 Terminal 3

Para este equipo, la configuración del archivo hosts debe ser:

```
192.168.0.3     ws003     ws003.localdomain     ws003
```

3.2.5.4 Hosts.allow, hosts.deny

En un servidor Linux el acceso a los servicios que usan wrappers TCP y que se basan en el nombre de la máquina, lo controlan dos archivos: /etc/hosts.allow y /etc/hosts.deny. Estos archivos usan un formato simple para controlar el acceso a los servicios.

Siempre y cuando no se haya especificado alguna regla en hosts.allow y hosts.deny, la regla por defecto consiste en permitir a cualquiera el acceso a los servicios.

El orden es importante ya que las reglas que se encuentran en hosts.allow tienen precedencia con respecto a las especificadas en hosts.deny. Aunque una regla deniegue el acceso a un determinado servicio en hosts.deny, los clientes que tengan el permiso para acceder al servicio en hosts.allow lo mantienen.

Las reglas de cada uno de los archivos están colocadas por orden de importancia por lo que hay que respetar dicho orden. Todos los cambios que se hagan en estos archivos afectan inmediatamente a los servicios a los que se refieran; no se requiere volver a arrancar los servicios. Todas las reglas del control de acceso que se encuentran en las líneas hosts.allow y hosts.deny, y los espacios en blanco o las líneas que empiezan con (#) no se toman en cuenta.

Cada regla tiene que estar en una línea separada.

Sintaxis:

```
<daemon_list>: <client_list>[: <shell_command> ]
```

Cada una de estas opciones se refiere a una parte específica de la regla:

- daemon_list: Es una colección de uno o varios nombres de procesos o de comodines especiales separados por espacios en blanco.

- client_list: Uno o varios nombres de máquinas, direcciones de máquina, funciones o comodines separados por espacios en blanco que se usan cuando el nombre de un determinado proceso se adapta a un determinado servicio.

- shell_command: Es un componente opcional que indica la realización de algo en el caso de que se use una regla.

Las características son de bastante ayuda cuando se especifican qué grupos de clientes pueden o no tener acceso a un determinado servicio. Así al poner el punto "." al comienzo de una cadena, todas las máquinas que se encuentran en esa cadena deben cumplir dicha regla. De tal manera que .domain.com engloba tanto a system1.domain.com como a system2.domain.com.

El punto "." al final de una cadena tiene el mismo efecto, excepto que el sentido cambia, ya que ésto se usa sobre todo para las direcciones IP ya que una regla relativa a 192.168.0. se aplica a todo el bloque C de las direcciones IP. Las expresiones de las máscaras de red también se pueden usar como característica para controlar el acceso a un determinado grupo de direcciones IP.

También puede usar asteriscos (*) o interrogaciones (?) para seleccionar grupos enteros de nombres de máquinas o de direcciones IP siempre y cuando no las use en la misma cadena de las otras características.

Si la lista de los nombres de máquinas que tienen acceso a un determinado servicio es muy larga o es difícil de controlar en los archivos host.allow o hosts.deny, puede especificar el recorrido completo a un archivo (como /etc/telnet.hosts.deny). Este archivo contiene diferentes nombres de máquinas, direcciones de máquinas o características separados por espacios en blanco a los que desea conceder o denegar el acceso a dicho servicio.

El método también funciona para compartir las listas de control del acceso entre varios servicios ya que los cambios que se quieran realizar solamente se tienen que hacer en un archivo para que afecten a cada servicio.

Los siguientes comodines se usan en las reglas para el control del acceso en vez de usar determinados nombres de máquinas o grupos de nombres de máquinas:

- ALL: Para permitir a un cliente tener acceso a todos los servicios, se utiliza ALL en la sección de los demonios.

- LOCAL: Hace corresponder todos los nombres de máquinas que no contengan un punto ".".

- KNOWN: Hace corresponder todas las máquinas cuyos nombres y direcciones se conozcan o en el caso en que se conozca el usuario.

- UNKNOWN: Hace corresponder todas las máquinas cuyos nombres y direcciones sean desconocidas o en el caso en el que se desconozca el usuario.

- PARANOID: Hace corresponder todas las máquinas cuyo nombre no se corresponda con la dirección.

El script de instalación del LTSP agregará algunas entradas en el archivo /etc/hosts.allow para permitir que dhcp, tftp y portmap funcionen correctamente. Estas entradas dan por hecho que se están usando direcciones IP de la clase C 192.168.0.0.

Si se utilizan direcciones diferentes se deben sustituir adecuadamente. Como el archivo hosts.allow tiene prioridad sobre el archivo hosts.deny, en realidad los cambios más importantes se efectúan en el primero.

El archivo hosts.allow tiene la siguiente estructura:

```
#
# hosts.allow    This file describes the names of the hosts which are
#                allowed to use the local INET services, as decided
#                by the '/usr/sbin/tcpd' server.
#

## LTS-begin ##

#
# The lines between the 'LTS-begin' and the 'LTS-end' were added
# on: sab sep 27 12:34:50 CDT 2003 by the ltsp installation script.
# For more information, visit the ltsp homepage
# at http://www.ltsp.org
#

bootpd:   0.0.0.0
in.tftpd: 192.168.0.
portmap:  192.168.0.

## LTS-end ##
```

Tabla 3-6 Ejemplo de configuración del archivo hosts.allow

Y el archivo hosts.deny, la siguiente:

```
#
# hosts.deny    This file describes the names of the hosts which are
#               *not* allowed to use the local INET services, as decided
#               by the '/usr/sbin/tcpd' server.
#
# The portmap line is redundant, but it is left to remind you that
# the new secure portmap uses hosts.deny and hosts.allow.  In particular
# you should know that NFS uses portmap!
```

Tabla 3-7 Ejemplo de configuración del archivo lts.conf

Estos archivos, permiten o niegan respectivamente algún servicio a las direcciones especificadas (en este caso las terminales), en algunas ocasiones pueden definirse direcciones IP concretas y en otros rangos de direcciones; incluso pueden combinarse.

Pero hay que tener cuidado al manipular estos archivos, ya que forman parte del firewall del sistema.

3.2.5.4.1 Terminal 1

Para este equipo, la configuración del archivo hosts.allow debe ser:

```
ALL:    192.168.0.1
```

En el archivo hosts.deny no es necesario hacer alguna configuración para esta terminal.

3.2.5.4.2 Terminal 2

Para este equipo, la configuración del archivo hosts.allow debe ser:

```
ALL:    192.168.0.2
```

En el archivo hosts.deny no es necesario hacer alguna configuración para esta terminal.

3.2.5.4.3 Terminal 3

Para este equipo, la configuración del archivo hosts.allow debe ser:

```
ALL:    192.168.0.3
```

En el archivo hosts.deny no es necesario hacer alguna configuración para esta terminal.

3.2.5.5 Exports

Es sencillo configurar un sistema para compartir archivos y directorios usando NFS, cada sistema de archivos que se exporta a usuarios remotos vía NFS, así como los derechos de acceso relativos a ellos, es localizado en el archivo /etc/exports. Este archivo es leído por el comando exportfs que da a rpc.mountd y rpc.nfsd la información necesaria para permitir el montaje remoto de un sistema de archivos por una máquina autorizada. Por omisión, rpc.mountd desaprueba el montaje de todos los directorios, lo cual es una medida de seguridad muy adecuada. Si se desea permitir montar un directorio de NFS, debe ser exportado, para lograr esto se debe especificar en el archivo exports.

Cada línea define un directorio y los anfitriones a los que se les permite montarlo. Un nombre de anfitrión es usualmente un nombre de dominio pero puede contener adicionalmente los comodines "*" y "?" los cuales funcionan igual que en el Bourne Shell de Linux. Por ejemplo, lab*.foo.com coincide con lab01.foo.com así como laboratory.foo.com. El anfitrión puede ser especificado también usando un rango de direcciones IP en la forma dirección/máscara de red.

Cuando se verifica a un anfitrión cliente contra el archivo exports, rpc.mountd busca el nombre del anfitrión cliente usando la llamada gethostbyaddr. Con DNS, esta llamada devuelve el nombre canónico del anfitrión cliente, así debe asegurarse de no usar alias en exports. En un entorno NIS el nombre devuelto es la primera coincidencia de la base de datos de anfitriones, y sin DNS o NIS, el nombre devuelto es el primer nombre de anfitrión encontrado en el archivo hosts que coincida con la dirección del cliente.

El nombre del anfitrión es seguido por una lista opcional de parámetros, los cuales pueden ser:

- secure: Asegura que la conexión se haga desde un puerto origen reservado, por ejemplo, uno que sea menor que 1024. Este parámetro es puesto por omisión.

- insecure: Este parámetro revierte el efecto de secure.

- ro: Provoca que el montaje de NFS sea para sólo lectura. Este parámetro está activado por omisión.

- rw: Esta opción monta la jerarquía de archivos en lectura-escritura.

- root_squash: Esta característica de seguridad deniega a los superusuarios en los hosts especificados cualquier derecho de acceso especial mapeando las peticiones desde el uid 0 en el cliente al uid 65534 (es decir, -2) en el servidor. Este uid debe ser asociado con el usuario nobody.

- no_root_squash: Esta opción no mapea las peticiones desde uid 0. Este parámetro está habilitado por omisión, así los superusuarios tienen acceso de supervisor a los directorios exportados de su sistema.

- link_relative: Convierte los enlaces simbólicos absolutos (donde el contenido del enlace comienza con un slash) en enlaces relativos. Esta opción sólo tiene sentido cuando está montado el sistema de archivos entero de un anfitrión; por otra parte, algunos de los enlaces podrían apuntar a ninguna parte, o peor aún, a archivos que nunca debieran apuntar. Esta opción está habilitada de forma predeterminada.

- link_absolute: Esta opción deja todos los enlaces simbólicos existentes.

- map_identity: Esta opción le indica al servidor asumir que el cliente usa el mismo uid y gid que el servidor. Este parámetro está habilitado por omisión.

- map_daemon: Indica al servidor de NFS asumir que el cliente y el servidor no comparten el mismo espacio uid/gid, rpc.nfsd entonces construye una lista que mapea los IDs entre cliente y servidor preguntando al demonio rpc.ugidd del cliente.

- map_static: Esta opción le permite especificar el nombre de un archivo que contiene un mapa estático de uids. Por ejemplo, map_static=/etc/nfs/vlight.map especificaría el archivo /etc/nfs/vlight.map como un mapa de uid/gid.

- map_nis: Esta opción causa que el servidor de NIS haga un mapeado de uid y gid.

- anonuid y anongid: Estas opciones le permiten especificar el uid y el gid de la cuenta anónima. Esto es útil si tiene un volumen exportado para montajes públicos.

En esencia, éste archivo define que directorios serán compartidos a otras máquinas, en el caso del LTSP, definirá los directorios a los cuales podrán tener acceso las terminales gráficas.

La estructura del archivo creada por los scripts del LTSP es la siguiente:

```
## LTS-begin ##
#
# The lines between the 'LTS-begin' and the 'LTS-end' were added
# on: Sat sep 27 12:34:50 CDT 2003 by the ltsp installation script.
# For more information, visit the ltsp homepage
# at http://www.ltsp.org
#

/opt/ltsp/i386              192.168.0.0/255.255.255.0(ro,no_root_squash,sync)
/var/opt/ltsp/swapfiles     192.168.0.0/255.255.255.0(rw,no_root_squash,async)

#
# The following entries need to be uncommented if you want
# Local App support in ltsp
#
#/home              192.168.0.0/255.255.255.0(rw,no_root_squash,sync)

## LTS-end ##
```

Tabla 3-8 Ejemplo de configuración del archivo exports

3.2.5.6 lts.conf

Este archivo contiene las configuraciones necesarias para crear el entorno LTSP; permite especificar los parámetros que las terminales requieren para poder funcionar correctamente. Este archivo tiene una sintaxis simple, que consiste de múltiples secciones. Hay una sección por defecto, llamada [default] y hay secciones adicionales para terminales individuales. Si todas las terminales son idénticas, entonces se deben especificar todas las opciones bajo la sección [Default]. Las terminales pueden ser identificadas por nombre de host, dirección IP o dirección MAC. A continuación se describen los parámetros principales que acepta el archivo lts.conf:

A) Opciones generales

- Comentarios: Los comentarios comienzan con un símbolo de número ('#') y continúan hasta el final de la línea.

- LTSP_BASEDIR: Indica dónde está ubicado el sistema de archivos raíz del LTSP; por defecto es en /opt/ltsp/.

- SERVER: Este es el servidor utilizado por XDM_SERVER, TELNET_HOST, XFS_SERVER y SYSLOG_HOST, si alguno de los anteriores no está especificado explícitamente. Si se tiene una computadora que está actuando como el servidor de

todo, entonces se puede especificar su dirección IP aquí y omitir el resto. Si ningún valor es especificado, se toma por defecto la dirección 192.168.0.254.

- SYSLOG_HOST: Si se requiere enviar los mensajes de sistema a una máquina distinta al servidor por defecto, entonces se tiene que especificar dicha máquina aquí. Si ningún parámetro es especificado se utilizará la entrada de SERVER.

- NFS_SERVER: Especifica la dirección IP del servidor desde donde el sistema de archivos /home es montado. Por defecto se utiliza la entrada SERVER.

- USE_NFS_SWAP: Esta opción habilita el uso de memoria SWAP sobre NFS. Por defecto es N.

- SWAPFILE_SIZE: Permite controlar el tamaño del archivo de intercambio. El tamaño por defecto es 64MB.

- SWAP_SERVER: El archivo de intercambio puede existir en cualquier servidor de la red. Se puede especificar la dirección IP de dicho servidor. Por defecto se toma el valor de NFS_SERVER.

- NFS_SWAPDIR: Especifica el directorio que es exportado vía NFS. Por defecto es /var/opt/ltsp/swapfiles. Es necesario que dicho directorio esté registrado en el archivo /etc/exports.

- TELNET_HOST: Si la terminal es configurada para tener una sesión en modo texto, entonces el valor de este parámetro será usado como el host donde se haga la sesión de telnet. Si ningún valor es especificado se utilizará la entrada de SERVER.

- DNS_SERVER: Usada para crear el archivo resolv.conf.

- SEARCH_DOMAIN: Usada para crear el archivo resolv.conf.

MODULE_01 hasta MODULE_10

10 módulos de kernel pueden ser cargados mediante estas entradas. Por ejemplo:

```
MODULE_01 = uart401.o
MODULE_02 = sb.o io=0x220 irq=5 dma=1
MODULE_03 = opl3.o
```

Si el valor de este parámetro es una ruta absoluta, entonces el comando insmod será utilizado. De otro modo, se utilizará modprobe.

- RAMDISK_SIZE: Cuando la terminal se inicia, crea un disco RAM y lo monta en el directorio /tmp. Se puede controlar el tamaño de este sistema de archivos con este parámetro. Se debe especificar las unidades en kilobytes (1024 bytes). Por ejemplo para crear un disco RAM de 2 MB se debe especificar RAMDISK_SIZE=2048.

- RCFILE_01 hasta RCFILE_10: Algunos scripts adicionales pueden ser ejecutados por rc.local. Se debe colocar el script en el directorio /etc/rc.d y especificar el nombre en alguna de estas entradas.

- SOUND: Si el paquete LTSP Sound está instalado, se necesita poner el valor 'Y' en esta entrada y ejecutar el script rc.sound para configurar la tarjeta de sonido y el demonio. Por defecto es N.

- RUNLEVEL: El runlevel determina el modo en el cual la terminal estará corriendo. Los siguientes runlevels son los soportados:

  3: Iniciará un shell. Útil para depuración de la terminal.
  4: Correrá una o más sesiones de telnet en el TELNET_HOST.
  5: Modo GUI. Arrancará X Window, y enviará una petición XDMCP al servidor, quien devolverá una pantalla de login a la terminal. Se necesita tener un Display Manager corriendo en el servidor, como XDM, GDM o KDM.

- TELNET_SESSIONS: Indica cuántas sesiones de telnet se pueden correr. Cada sesión estará en una diferente pantalla virtual, a las que se puede acceder con las teclas ALT-F1 a ALT-F9. El valor por defecto es 2.

B) Opciones de X Window

- XDM_SERVER: Si se requiere que XDM corra en una máquina que no sea el servidor por defecto, entonces se debe especificar aquí. Si este parámetro no es especificado, entonces se usará la entrada en SERVER.

- XSERVER: Esta entrada define qué servidor X correrá la terminal. Para placas PCI y AGP, este parámetro no debería ser necesario. El script rc.local debería autodetectar la tarjeta. También se puede activar auto a fin de indicar que se tratará de autodetectar el video. Para placas ISA, o para especificar un servidor X en particular, se puede poner el nombre del driver o servidor X. Si el valor comienza con 'XF86_', entonces XFree 3.3.6 será utilizado, sino, XFree 4.1.x lo será. El valor por defecto es auto.

- X_MODE_0 hasta X_MODE_2 : Hasta 3 resoluciones pueden ser configuradas para la terminal. Esta entrada puede tomar dos tipos diferentes de valores. Puede ser tanto una resolución como un modeline completo:

X_MODE_0 = 800x600 o bien

X_MODE_0 = 800x600 60.75 800 864 928 1088 600 616 621 657 -HSync –Vsync

Si ninguna de las entradas X_MODE_x es especificada, entonces se utilizarán los modelines por defecto, y las resoluciones serán de 1024x768, 800x600 y 640x480. Si una o más entradas X_MODE_x es especificada, entonces no se tendrán en consideración los modelines por defecto.

- X_MOUSE_PROTOCOL: Cualquier valor que funcione con el protocolo de puntero XFree86 puede ser especificado. Los valores típicos incluyen 'Microsoft' y 'PS/2'. El valor por defecto es 'PS/2'.

- X_MOUSE_DEVICE: Este es el dispositivo al cual el mouse está conectado. Si es un mouse serial, debería ser un puerto serie, como /dev/ttyS0 o /dev/ttyS1. Si es un mouse PS/2, este valor debería ser /dev/psaux. El valor por defecto es /dev/psaux.

- X_MOUSE_RESOLUTION: Este es el valor 'Resolution' que se encuentra en cualquier archivo XF86Config. Un valor típico para un mouse serial es 50 y para uno PS/2 es 400. El valor por defecto es 400.

- X_BUTTONS: Indica al sistema cuántos botones tiene el mouse, usualmente son 2 o 3 y el valor por defecto es 3.

- X_MOUSE_EMULATE3BTN: Esto le indica al servidor X que debe emular el tercer botón del mouse cuando se presionen simultáneamente los botones izquierdo y derecho. El valor por defecto es N.

- X_MOUSE_BAUD: Para ratones seriales, define la tasa de baudios. El valor por defecto es 1200.

- X_COLOR_DEPTH: Este es el número de bits a utilizar para la profundidad del color. Los valores posibles son 8, 15, 16, 24 y 32. 8 bits darán 256 colores, 16 - 65536, 24 - 16 millones y 32 - 4,2 billones de colores. No todos los servidores X soportan estos valores. El valor por defecto es 16.

- USE_XFS: Se tiene la opción de correr un Servidor de Fuentes X (XFS, X Font Server) o bien leer las fuentes vía NFS. El servidor de fuentes debería proveer un camino simple para mantener todas las fuentes en un solo lugar, pero hay algunos problemas cuando el número de terminales supera las 40. Los 2 valores en esta opción son Y o N, el valor por defecto es N. Si se desea usar un Servidor de Fuentes, entonces se debe utilizar la entrada XFS_SERVER para especificar su dirección IP.

- XFS_SERVER: Si se está usando un Servidor de Fuentes, se debe especificar su dirección IP. Si no está especificado, entonces se usará por defecto la entrada especificada en SERVER.

- X_HORZSYNC: Define la sincronización horizontal del monitor y el valor por defecto es "31-62".

- X_VERTREFRESH: Define con que frecuencia se refresca verticalmente el monitor. Por defecto es "55-90".

- XF86CONFIG_FILE: Si se desea tener un archivo propio de configuración XF86Config, se puede especificar en esta opción. Sólo se tiene que colocar en el directorio /opt/ltsp/i386/etc. No importa su nombre, siempre y cuando dicho nombre sea indicado en éste parámetro. Por ejemplo:

- XF86CONFIG_FILE = XF86Config.ws004: Los scripts de configuración del LTSP crean este archivo el cual tiene la siguiente estructura:

```
    # Keyboards
    XkbSymboles     = "us(pc101)"
    XkbModel        = "pc101"
    XkbLayout       = "us"

    USE_XFS         = N
    LOCAL_APPS      = N
    RUNLEVEL        = 5

    # uncomment the following line to enable floppy support
    #RCFILE_01      = floppyd

    # uncomment the following line to enable USB support
    #RCFILE_02      = usb

    # enable sound by default
    SOUND           = Y

    # default sound volume
    VOLUME          = 75

    ### For ISA sound cards, you have to specify the module to use:
    # SMODULE_01    = sb io=0x220 irq=5 dma=1
#-----------------------------------------------------------------------
# Below are sample mode lines for a variety of vertical refresh rates and
# resolutions.  They are used to define the default client screen resolution.
# Some lines may not work with a given monitor and video chipset.
# To avoid damaging a monitor and video card, only specify mode lines that
# your client's hardware can support.
# Uncomment only *one* of the following X_MODE_0 lines at a time, or add one
# of your own.
```

```
# 60 Hz Resolutions
#  X_MODE_0 = 640x480  25.175  640  656  752  800   480 490 492 525  -hsync -vsync
#  X_MODE_0 = 800x600  40      800  840  968 1056   600 601 605 628  +hsync +vsync
#  X_MODE_0 = 1024x768 65     1024 1048 1184 1344   768 771 777 806  -hsync -vsync

# 70 Hz Resolutions (Use instead of 72 Hz for 1024x768)
#  X_MODE_0 = 1024x768 75     1024 1048 1184 1328   768 771 777 806  -hsync -vsync

# 72 Hz Resolutions
#  X_MODE_0 = 640x480  31.5    640  664  704  832   480 489 492 520  -hsync -vsync
#  X_MODE_0 = 800x600  50      800  856  976 1040   600 637 643 666  +hsync +vsync
#  X_MODE_0 = 1024x768 75     1024 1048 1192 1296   768 771 777 806  -hsync -vsync

# 75 Hz Resolutions
#  X_MODE_0 = 800x600  49.5    800  816  896 1056   600 601 604 625  +hsync +vsync

# 85 Hz Resolutions
#  X_MODE_0 = 800x600  60.75   800  864  928 1088   600 616 621 657  -hsync -vsync

#----------------------------------------------------------------------
#
# Example of specifying X settings for a workstation
#
[ws001]
    XSERVER         = auto
    LOCAL_APPS      = N
    USE_NFS_SWAP    = N
    SWAPFILE_SIZE   = 48m
    RUNLEVEL        = 5

[ws002]
    XSERVER         = XF86_SVGA
    LOCAL_APPS      = N
    USE_NFS_SWAP    = N
    SWAPFILE_SIZE   = 64m
    RUNLEVEL        = 3

#
# ws004 is my virtual workstation running in a VMware session
#
[ws004]
    DNS_SERVER      = 192.168.0.254
    XSERVER         = auto
    X4_BUSID        = "PCI:0:15:0"
    X_MODE_0        = 800x600
    LOCAL_APPS      = N
    USE_NFS_SWAP    = N
    SWAPFILE_SIZE   = 64m
    RUNLEVEL        = 5

#----------------------------------------------------------------------
#
# Example of a workstation configured to load some modules
```

```
#
#[ws001]
#   MODULE_01      = agpgart.o          # This is for i810 video
#   MODULE_02      = uart401.o
#   MODULE_03      = sb.o io=0x220 irq=5 dma=1
#   MODULE_04      = opl3.o

#-----------------------------------------------------------------------
#
# Example of ws001 configured for local apps
#
#[ws001]
#   LOCAL_APPS     = Y
#   LOCAL_WM       = Y
#   NIS_DOMAIN     = ltsp
#   NIS_SERVER     = 192.168.0.254

#-----------------------------------------------------------------------
#
# Example of a serial printer attached to /dev/ttyS1 on workstation ws001
#
#[ws001]
#   PRINTER_0_DEVICE  = /dev/ttyS1
#   PRINTER_0_TYPE    = S          # P-Parallel, S-Serial
#   PRINTER_0_PORT    = 9100       # tcp/ip port: defaults to 9100
#   PRINTER_0_SPEED   = 9600       # baud rate: defaults to 9600
#   PRINTER_0_FLOWCTRL = S         # Flow control: S-Software (XON/XOFF),
#                        # H-Hardware (CTS/RTS)
#   PRINTER_0_PARITY  = N          # Parity: N-None, E-Even, O-Odd
#                        # (defaults to 'N')
#   PRINTER_0_DATABITS = 8         # Databits: 5,6,7,8 (defaults to 8)
```

Tabla 3-9 Ejemplo de configuración del archivo lts.conf

3.2.5.6.1 Terminal 1

Para este equipo, la configuración del archivo lts.conf debe ser:

```
[ws001]
        XSERVER             = Xvesa
        LOCAL_APPS          = N
        USE_NFS_SWAP        = N
        SWAPFILE_SIZE       = 48m
        RUNLEVEL            = 5
        X_HORZSYNC          = "30.9-49.0"
        X_VERTREFRESH       = "50.0-100.0"
        X_MODE_0            = 800x600
```

3.2.5.6.2 Terminal 2

Para este equipo, la configuración del archivo lts.conf debe ser:

```
[ws002]
    XSERVER          = Xvesa
    LOCAL_APPS       = N
    USE_NFS_SWAP     = N
    SWAPFILE_SIZE    = 48m
    RUNLEVEL         = 5
    X_HORZSYNC       = "30.9-49.0"
    X_VERTREFRESH    = "50.0-100.0"
    X_MODE_0         = 800x600
```

3.2.5.6.3 Terminal 3

Para este equipo, la configuración del archivo lts.conf debe ser:

```
[ws003]
    XSERVER          = Xvesa
    LOCAL_APPS       = N
    USE_NFS_SWAP     = N
    SWAPFILE_SIZE    = 48m
    RUNLEVEL         = 5
    X_MODE_0         = 800x600
```

3.3 Terminales

Una vez que se haya terminado de instalar y configurar todos los paquetes necesarios en el servidor, el siguiente paso es crear una imagen de sistema operativo para que las terminales puedan arrancar. Donde las posibilidades de arranque podrían ser las siguientes:

- Arranque desde la tarjeta de red: Las tarjetas necesitan un chip EPROM para grabar en él la imagen de arranque. Arranque desde la propia BIOS si se incorpora PXE (Preboot-eXecution Environment).

- Arranque desde disquete: La imagen de arranque se graba en un disquete 3 ½ " de alta densidad. Esta es la solución más económica y sencilla.

- Arranque desde el disco duro: Configurando el archivo Config.sys y Autoexec.bat en WINDOWS.

3.3.1 Creación del disquete de arranque

Es importante mencionar que no basta con crear una imagen de arranque común, la diferencia radica en que ésta deberá estar etiquetada especialmente para que el kernel permita arrancar la terminal vía red.

La forma más común es bajar la imagen de arranque y copiarla en un disquete[13]. Se deberá elegir la opción Floppy Bootable Rom Image, lo cual generará una imagen que se puede escribir en un disquete con los comandos dd o cat, desde Linux; o rawrite en MS-DOS.

Este disquete primero intentará detectar la tarjeta de red y luego iniciará el proceso de arranque como si el código estuviera en una eprom desde la misma tarjeta de red. Por ejemplo, para generar el disquete de arranque para la tarjeta MX98715 de la terminal 3, se tiene que hacer lo siguiente:

- Ir al sitio y hacer click en la liga de la versión 5.0.7.
- Seleccionar el tipo de tarjeta de red, en este caso es "mx98715".
- Elegir el formato de salida Floppy Bootable ROM Image (.lzdsk).
- Hacer click en "Get ROM" para generar la imagen de arranque. Se creará el siguiente archivo:

```
eb-5[1].0.7-mx98715.lzdsk
```

Para crear el disco de arranque no basta con copiar la imagen al disquete, es necesario que éste sea booteable para esto se utiliza cualquiera de los siguientes comandos:

En Linux se utiliza:

```
cat eb-5[1].0.7-mx98715.lzdsk > /dev/fd0
```

Esto copiará la imagen en el primer sector del disquete.

[13] Página del sitio para descargar la imagen de arranque http://www.rom-o-matic.net

O desde MS-DOS, usando el comando rawrite, el cual viene incluido en el primer disco de instalación de Fedora Core, en la carpeta DOSUTILS.

Se escribe el comando:

```
D:\DOSUTILS\rawrite
```

Se pedirá la ruta de la imagen de arranque:

```
C:\eb-5[1].0.7-mx98715.lzdsk
```

Y después la ruta donde se escribirá la imagen de arranque, en este caso es el disquete:

```
A:\
```

Este comando no acepta archivos con extensiones mayores a tres caracteres, por lo tanto se tendrá que cambiar la extensión del archivo, en este caso será .img.

```
C:\ren eb-5[1].0.7-mx98715.lzdsk   mx98715.img
```

Y se tendrá que repetir el proceso con el comando rawrite hasta que se haya logrado grabar correctamente la imagen en el disquete.

Después de crear el disco de arranque, será necesario probar que éste reconoce la tarjeta de red de la terminal, lo cual se puede comprobar si al bootear la terminal con el disquete se muestra la MAC address de la tarjeta y aparece un mensaje que indica que se está buscando un servidor DHCP.

3.3.2 Arranque por medio del disco duro

Otra alternativa para el arranque de las terminales es la creación de un menú en Windows en el disco duro local, lo cual permite iniciar Linux sin la necesidad de utilizar un disquete de arranque y a su vez iniciar Windows normalmente.

Es importante mencionar que esté método solo funciona en Windows 95 y 98, ya que en versiones posteriores el soporte para el menú del archivo config.sys fue

deshabilitado. Las terminales 1 y 2 tienen instalado Windows 95 por lo que utilizarán este método, en cambio la terminal 3 tiene instalado Windows Millenium, por lo que se utilizará un disquete de arranque.

Para crear el menú se requiere hacer lo siguiente:

- Ir al sitio[14] y hacer click en la liga de la versión 5.0.7.
- Seleccionar el tipo de tarjeta de red, en este caso es "3c509" para las terminales 1 y 2.
- Elegir el formato de salida DOS Executable ROM Image (.com).
- Hacer click en "Configure" para definir que en el proceso de arranque se muestre una barra giratoria en lugar de una sucesión de puntos.
- Hacer click en "Get ROM" para generar la imagen de arranque. Se creará el siguiente archivo:

```
eb-5[1].0.7-3c509.com
```

Se debe renombrar el archivo generado con un nombre descriptivo, no mayor a 8 caracteres. Esto se puede hacer directamente en el explorador de Windows o en la línea de comandos de MS-DOS.

```
C:\ren  eb-5[1].0.7-3c509.com   3c509.com
```

Posteriormente se debe crear un directorio directamente en C:\ llamado ltsp, donde se colocará el archivo 3c509.com.

```
C:\md ltsp
C:\cp 3c509.com  ltsp\
```

Se tiene que editar el archivo config.sys y agregar lo siguiente al inicio:

[14] Página del sitio para crear al menú de inicio http://www.rom-o-matic.net

```
[Menu]
MenuItem=windows,Microsoft Windows 95/98
MenuItem=linux,Linux Terminal Server Project
MenuDefault=windows,10

[linux]
DOS=NOAUTO

[windows]
```

Esto creará el menú con las opciones de inicio Windows y Linux.

También se tiene que editar el archivo autoexec.bat agregando lo siguiente al inicio:

```
@ECHO OFF
GOTO %CONFIG%

:linux
C:\ltsp\3c509.com

:windows
```

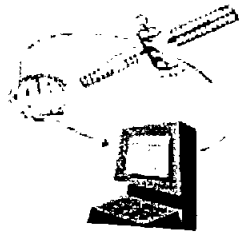
Esto permitirá realmente ejecutar la imagen de arranque una vez que se haya seleccionado la opción Linux en el menú de inicio.

Hasta éste punto preparamos el equipo para instalar adecuadamente el sistema operativo así como los diferentes componentes que intervienen en la implementación; fueron instalados los archivos de sistema y aprendimos a crear el disco de arranque. Ahora estamos listos para iniciar todo el sistema, analizaremos el inicio de una terminal para lograr detectar problemas y con lo que hemos visto en los capítulos anteriores podremos instalar sin problemas el software adicional así como la configuración a internet.

PUESTA EN MARCHA

OBJETIVO ESPECÍFICO

Aplicar los conocimientos teóricos y prácticos para garantizar el buen funcionamiento del servidor de terminales gráficas en el momento de iniciar todo el sistema. Entender el proceso de cómo inicia una terminal para poder detectar y darle la solución más óptima a los posibles problemas.



4

INTRODUCCIÓN

Hasta este punto se han instalado los paquetes del LTSP y del software necesario, también se han realizado las configuraciones correspondientes a los archivos de sistema; sin embargo falta comprobar que efectivamente cada una de las terminales inicien una sesión gráfica debido a que estas configuraciones no garantizan que dichas terminales funcionen correctamente, para ello se cuenta con un apartado de detección y solución de problemas más comunes.

Una vez que han sido solucionados los problemas y se han iniciado las terminales gráficamente estarán listas para ser conectadas en red de forma local y a su vez tendrán acceso a la red global.

Tomando en cuenta que en el laboratorio se utiliza software comercial de electrónica, se hizo una recopilación de software de este tipo, pero que puede correr bajo Linux y con la ventaja de que es libre.

4.1 Teoría de operación

El arranque de una terminal gráfica aplicando el LTSP involucra muchos pasos, es importante conocer y comprender de forma general cada uno de ellos; ya que si en algún momento surgen problemas, estos podrían solucionarse más rápidamente, debido a que se puede deducir el origen de ellos.

El proceso de arranque de una terminal aplicando el LTSP es el siguiente:

- Cuando se enciende la terminal, ésta busca entre los dispositivos de almacenamiento disponibles un sistema operativo que iniciar; cuando detecta el código de arranque, éste se empezará a ejecutar.

- El código de inicio tratará de detectar una tarjeta de red y una vez que lo hace, la inicializa.

- Se hará una petición de broadcast buscando un servidor dhcp en la red local, esta petición incluirá la dirección MAC de la tarjeta de red.

- El proceso inetd del servidor verá la petición de broadcast e invocará al demonio dhcpd para responderla.

- El demonio dhcpd leerá el archivo de configuración, dhcpd.conf, y tratará de localizar la entrada que corresponda con la dirección MAC de la terminal. Al ser encontrada, la pondrá en un paquete de respuesta y se lo enviará a la terminal que pidió la información. Ciertos parámetros serán incluidos en este paquete, siendo los más importantes:

    - Dirección IP asignada a la terminal.
    - Máscara de subred de la red de área local.
    - Directorio home del archivo de inicio.
    - Nombre del kernel a bajar.

- El código de inicio recibirá el paquete de respuesta dhcp y configurará la interfaz TCP/IP en la tarjeta de red con los parámetros proporcionados.

- El código de inicio enviará una petición TFTP al servidor para empezar a bajar el kernel.

- Una vez que el kernel fue completamente bajado por la terminal, éste tomará el control.

- El kernel empezará a ejecutarse, inicializando todo el sistema y sus periféricos.

- El código de inicio le puede pasar información al kernel y lo hace construyendo una línea de comandos, similar a la manera en que LILO o GRUB le pasa parámetros.

- Si el kernel necesita hacer una petición dhcp, el servidor responderá con otro paquete, conteniendo la información que el kernel necesita para continuar. Los parámetros importantes en esta respuesta son:

    - Dirección IP asignada a la terminal.
    - Máscara de subred de la red de área local.
    - El directorio raíz a ser montado vía NFS.
    - La puerta de enlace (gateway).
    - El servidor de nombres (DNS).
    - El nombre de host de la terminal.

- El directorio raíz de la terminal será montado vía NFS. Este sistema de archivos será de sólo lectura; porque se pueden tener varias terminales montando el mismo sistema de archivos, y con esto se evita que en alguna de ellas se modifiquen los contenidos del mismo.

- En este punto, el control será pasado del kernel al proceso init.

- Init leerá el archivo inittab y empezará a configurar el ambiente.

- Uno de los primeros elementos de inittab es el archivo rc.local, que correrá mientras la terminal está en el estado 'sysinit'.

- El script en rc.local creará un disco RAM de 1 MB que contendrá todas las cosas que necesitan ser escritas o modificadas de alguna manera.

- Este disco RAM será montado en el directorio /tmp. Cualquier archivo que necesite ser escrito estará en el directorio /tmp, con un enlace simbólico apuntando hacia él.

- El directorio /proc es montado.

- La interfaz de loopback es configurada.

- Varios directorios son creados dentro de /tmp para que contengan algunos de los archivos transitorios que son necesarios mientras el sistema está corriendo. Algunos de ellos son:

    - /tmp/compiled
    - /tmp/var
    - /tmp/var/run

- /tmp/var/log
- /tmp/var/lock
- /tmp/var/lock/subsys

- El archivo XF86Config es generado y está basado en las entradas del archivo lts.conf. Aquí es donde la información acerca del tipo de mouse, y otros parámetros para el servidor X son combinados para crear el archivo de configuración.

- El script start_ws es creado, éste es responsable de iniciar el servidor X y cuando lo inicia, es dirigido para enviar peticiones XDMCP al servidor XDM. Esto funciona basado en la información encontrada en el archivo lts.conf.

- El archivo syslog.conf es creado, éste archivo contiene información que le indica al demonio syslogd a qué host de la red le deberá enviar la información de log. Este host de syslog está especificado en el archivo lts.conf. Hay un enlace simbólico llamado /etc/syslog.conf que apunta a /tmp/syslog.conf.

- El demonio syslogd se inicia, usando el archivo de configuración recientemente creado.

- El control es pasado otra vez a init y buscará en la entrada initdefault para saber a cuál runlevel entrar, por lo general el runlevel por defecto es 2.

- El runlevel 2 hará que se ejecute el script set_runlevel, que leerá el archivo lts.conf para saber en qué modo correr la terminal.

- Los niveles de ejecución estándar de LTSP son 3, 4 y 5.

  - El nivel 3 iniciará en un shell en la terminal.
  - El nivel 4 ejecutará una o más sesiones de telnet en el servidor. Esto es ideal para remplazar viejas terminales de tipo serial.
  - El nivel 5 entrará en el modo GUI, levantando el cliente X y haciendo una petición XDMCP al servidor, quien mandará una ventana de diálogo de inicio para permitir al usuario autenticarse. Hace falta un Display Manager corriendo en el servidor, como XDM, GDM o KDM.

4.2 Iniciando las terminales

La estrategia que se adoptó fue probar de manera independiente cada una de las terminales, para lo cual es necesario conectarlas directamente al servidor por medio de un cable cruzado; la ventaja de hacerlo de esta forma es que se tiene un mejor control en cuanto a la posible detección y solución de problemas.

Una vez que se haya comprobado que todas las terminales inician correctamente el sistema operativo y una respectiva sesión gráfica, se podrá implementar realmente la red, es decir; las terminales se conectarán al switch y éste a su vez al servidor, para lo cual será necesario utilizar cable directo en lugar de cable cruzado.

Asumiendo que las configuraciones del servidor fueron las adecuadas, al iniciar una sesión será desplegada la siguiente pantalla:

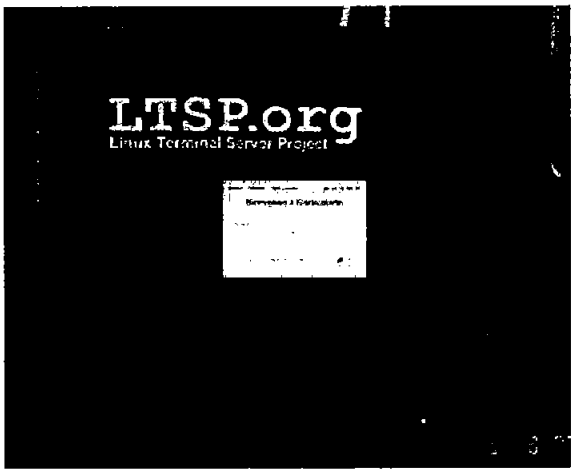


Figura 4-1 Pantalla de bienvenida del LTSP

Debido a su importancia, se describirá en la siguiente sección los problemas que se detectaron así como otros posibles errores en la implementación del LTSP, además la forma en que pueden ser solucionados.

4.3 Detección y solución de problemas

Los scripts de inicialización del LTSP hacen un gran trabajo al realizar ciertas configuraciones en los archivos necesarios, pero debido a la gran diversidad de hardware disponible para las terminales, es poco probable que éstas inicien completamente y de manera correcta la primera vez; es por eso que se debe comenzar con un proceso de detección y solución de problemas.

Conforme se va teniendo más experiencia en el proyecto LTSP, resulta más sencillo detectar los posibles errores y determinar la solución apropiada en cada caso; para ello es necesario observar hasta que punto ha llegado la terminal en su proceso de inicio.

A continuación se presentan los problemas más comunes, la manera de cómo detectarlos y sus posibles soluciones.

4.3.1 Conexiones

Es probable que la terminal no esté conectada apropiadamente a la red local, sobre todo cuando apenas se está implementando dicha red y ni siquiera se ha probado la conectividad entre las máquinas.

Para verificar esto, hay que observar que las luces de la tarjeta de red estén encendidas así como para todas las conexiones de red. Esto garantiza que al menos existe la conectividad entre la terminal y el dispositivo de la red local correspondiente.

En algunas ocasiones por cuestiones de prueba, es conveniente trabajar con la terminal conectada directamente al servidor; para ello es necesario que entre la terminal y el servidor (sin hacer uso de un hub o switch) se esté utilizando un cable cruzado.

Si se está utilizando un hub o switch, entonces hay que asegurarse que se está usando un cable directo, tanto entre las terminales y el hub, como entre el hub y el servidor.

En el capítulo 2 se muestra la configuración de los pines para construir un cable directo y un cable cruzado.

4.3.2 Imagen de arranque

La imagen de arranque debe reconocer la tarjeta de red de la terminal para poderla inicializar y configurar correctamente; si esta imagen no corresponde a la tarjeta de red, entonces no servirá para inicializar dicha terminal.

Cuando se inicia la terminal por medio de la imagen de arranque, se debe observar una salida como la siguiente:

```
loaded ROM segment 0x0800 length 0x4000 reloc 0x9400
Etherboot 5.0.1 (GPL) Tagged ELF for [LANCE/PCI]
Found AMD Lance/PCI at 0x1000, ROM address 0x0000
Probing...[LANCE/PCI] PCnet/PCI-II 79C970A base 0x1000, addr 00:50:56:81:00:01
Searching for server (DHCP)...
<sleep>
```

Tabla 4-1 Mensaje de salida cuando la imagen de arranque es apropiada

Para identificar si la imagen de arranque es la apropiada, ésta debe mostrar la dirección MAC de la tarjeta de red, sin embargo esto no garantiza que funcione completamente.

Si, por el contrario, se obtiene una salida como la siguiente, entonces es probable que la imagen de arranque generada no sea la apropiada para la tarjeta de red ya que no detecta ni muestra la dirección MAC.

```
ROM segment 0x0800 length 0x8000 reloc 0x9400
Etherboot 5.0.2 (GPL) Tagged ELF for [Tulip]
Probing...[Tulip]No adapter found
<sleep>
<abort>
```

Tabla 4-2 Mensaje de salida al iniciar la terminal con la imagen de arranque

Si no se muestran estos mensajes indicando que Etherboot a iniciado, entonces puede ser que el disquete esté dañado o que se haya grabado la imagen de arranque de manera incorrecta.

En el capítulo 3 se indica la manera de generar la imagen de arranque, así como la forma de grabarla en el disquete.

4.3.3 Problemas con el servicio DHCP

Una vez que la tarjeta de red es inicializada, el código de la imagen de arranque enviará una petición de broadcast a través de la red local, buscando algún servidor DHCP. Si la terminal obtiene una respuesta válida de parte del servidor DHCP, deberá configurar la tarjeta de red.

Si el servidor DHCP está funcionando correctamente, éste responderá a la terminal enviándole un paquete con la información que está definida en su archivo de configuración (dhcpd.conf) y que corresponde a la terminal que envió la petición.

```
ROM segment 0x0800 length 0x4000 reloc 0x9400
Etherboot 5.0.1 (GPL) Tagged ELF for [LANCE/PCI]
Found AMD Lance/PCI at 0x1000, ROM address 0x0000
Probing...[LANCE/PCI] PCnet/PCI-II 79C970A base 0x1000, addr 00:50:56:81:00:01
Searching for server (DHCP)...
<sleep>
Me: 192.168.0.1, Server: 192.168.0.254, Gateway 192.168.0.254
```

Tabla 4-3 Mensaje de que el servidor DHCP funciona correctamente

Si se muestra la línea que comienza con 'Me:', seguida por una dirección IP, entonces es un hecho que DHCP está funcionando correctamente. En ésta línea se mostrará la dirección IP asignada a la terminal, y las direcciones IP del servidor y del gateway.

Si por el contrario se muestra el siguiente mensaje en la terminal y no se obtiene respuesta por parte del servidor, es probable que haya un problema con el servicio DHCP.

```
Searching for server (DHCP)...
<sleep> <sleep>
```

Tabla 4-4 Mensaje de que quizá no funciona adecuadamente el servicio DHCP

Hay que notar que el mensaje <sleep> se puede mostrar por algunos segundos, debido al tráfico en la red, o porque por alguna razón el servidor estaba muy ocupado y no había podido atender la petición de la terminal; sin embargo no es normal que el servidor tarde mucho tiempo en dar una respuesta.

En el caso de que el servidor DHCP no responda, se tiene que averiguar las posibles causas; entre las cuales se encuentran las siguientes.

4.3.3.1 El demonio dhcpd

Para determinar si el demonio dhcpd está corriendo en el servidor se pueden ejecutar algunos comandos, e interpretar su salida; dhcpd normalmente corre en modo background escuchando en el puerto udp 67.

Se debe ejecutar el comando netstat para ver si hay algo escuchando en dicho puerto:

```
netstat -an | grep ":67 "
```

Se debe mostrar una salida similar a la siguiente:

```
udp    0      0      0.0.0.0:67      0.0.0.0:*
```

Tabla 4-5 Mensaje para saber si el demonio dhcpd esta corriendo en el servidor

La cuarta columna contiene la dirección IP y el puerto, separadas por dos puntos ":". Una dirección con solo ceros "0.0.0.0" indica que se está escuchando en todas las

interfases. Esto significa que se pueden tener varias tarjetas de red en el servidor; una interfaz eth0, otra eth1, etc., y que dhcpd esté escuchando en todas ellas.

Aunque se haya mostrado que existe un servicio escuchando en el puerto udp 67, esto no significa que efectivamente dhcpd sea el que lo esté haciendo; ya que BOOTP el cual es un servicio similar a DHCP, podría ser el que estuviera ocupando el puerto udp 67.

Actualmente en la mayoría de las distribuciones Linux solo se incluye el servicio DHCP, pero para asegurarse que el demonio esté ejecutándose realmente en el servidor, se tiene que teclear el siguiente comando:

```
ps aux | grep dhcpd
```

La salida es parecida a la siguiente:

```
root 23814 0.0 0.3 1676 820 ? S 15:13 0:00 /usr/sbin/dhcpd
root 23834 0.0 0.2 1552 600 pts/0 S 15:52 0:00 grep dhcp
```

Tabla 4-6 Mensaje para asegurarse que el demonio dhcpd esté ejecutándose

En realidad la primera línea es la que muestra que dhcpd está corriendo, la segunda es tan sólo el comando grep que se utilizó para hacer la búsqueda en la salida del comando ps.

Si no se muestra ninguna línea indicando que dhcpd está corriendo, entonces se necesita verificar que el servidor esté configurado para el runlevel 5 y que dhcpd esté listo para iniciar en dicho nivel. En sistemas basados en Fedora Core, se puede ejecutar el programa ntsysv para asegurar que el servicio DHCP esté seleccionado en el nivel de ejecución apropiado.

Existe una alternativa al comando ps, para saber si se está ejecutando dhcpd; se debe teclear lo siguiente:

```
service dhcpd status
```

Este comando mostrará el estado del servicio (si está detenido o en ejecución); si está ejecutándose indicará los ID de proceso que está utilizando.

```
Se está ejecutando dhcpd (pid 3693)...
```

Tabla 4-7 Mensaje que muestra el estado del servicio dhcpd

Si efectivamente el demonio dhcpd está detenido o con algún problema; se puede probar iniciando o reiniciando el servicio con estos comandos:

```
service dhcpd start
service dhcpd restart
```

En algunas ocasiones cuando se intenta iniciar dhcpd, aparece el siguiente mensaje:

```
st decls to leases file.
Wrote 0 new dynamic host decls to leases file.
Wrote 0 leases to leases file.
Not configured to listen on any interfaces!
```

Tabla 4-8 Mensaje de error de que no están activas las tarjetas de red

Este mensaje de error significa que no existen interfaces ethernet en el servidor o que no están activadas, esto se soluciona con el programa de configuración apropiado; en el caso de Fedora Core existe una herramienta llamada redhat-config-network, con ella se pueden configurar y activar las tarjetas de red del sistema.

Una vez que se haya(n) configurado la(s) tarjeta(s) de red, proporcionando su dirección IP, máscara de red, puertas de enlace, etc., y después de haberla(s) activado; se puede comprobar su estado con el comando ifconfig, el cual mostrará las interfaces ethernet activadas. En este momento, podrá iniciarse el demonio dhcpd sin ningún problema.

Si los comandos start o restart no arrojan errores, significa que dhcpd ahora sí, estará corriendo en el servidor.

4.3.3.2 Verificar la configuración de dhcpd.conf

Como se vio en el capítulo 3, el archivo dhcpd.conf contiene las configuraciones necesarias para que el servicio DHCP funcione correctamente; en ocasiones basta con adaptar ciertas cosas al archivo de ejemplo dhcpd.conf.example; particularmente las direcciones MAC de las terminales. Sin embargo hay situaciones en las que se debe modificar otros parámetros.

En primer lugar se debe verificar que la siguiente línea se encuentre antes del parámetro default-lease-time:

ddns-update-style none;

Esta línea es requerida a partir de la versión 3.0 de DHCP.

Hasta el momento en que la terminal necesita recibir la información para configurar la tarjeta de red y antes de comenzar a bajar el kernel desde el servidor, solo son requeridos ciertos parámetros del archivo dhcpd.conf.

Es necesario especificar las direcciones IP del servidor LTSP, del gateway y del servidor de dominio; en ambientes LTSP generalmente se utiliza un solo servidor que realice estas tres funciones, es decir una misma dirección IP será indicada en estos parámetros.

También se debe verificar que la entrada "hardware ethernet" concuerda exactamente con la dirección MAC de la terminal; ésta dirección se puede conocer cuando se inicia la terminal con la imagen de arranque apropiada.

hardware ethernet 00:E0:06:E8:00:84;

Finalmente se tienen que indicar las direcciones IP que serán asignadas a cada una de las terminales, estas se establecen en el parámetro "fixed-address".

fixed-address 192.168.0.1;

Con esta información bastará para que la terminal pueda configurar la tarjeta de red y comience a bajar la imagen del kernel desde el servidor LTSP.

4.3.3.3 Firewalls

Linux cuenta con herramientas llamadas firewalls, que disminuyen el riesgo de que algún intruso pueda comprometer la seguridad del sistema, sin embargo si estas no son configuradas adecuadamente para funcionar en conjunto con el ambiente LTSP, pueden ocasionar que el servidor no trabaje como se espera.

- Ipchains: Los kernels anteriores al 2.4 tenían la posibilidad de manipular la entrada y salida de los paquetes de datos usando ipchains, el cual utilizaba listas de reglas que se aplicaban a los paquetes en cada paso del proceso de filtrado.

Si en el servidor LTSP, se trabaja con un kernel cuya versión sea menor a la 2.4; se debe teclear el siguiente comando para determinar si ipchains está bloqueando la salida o entrada de paquetes:

```
ipchains -L –v
```

La salida debe ser similar a la siguiente:

```
Chain input (policy ACCEPT: 229714 packets, 115477216 bytes):
Chain forward (policy ACCEPT: 10 packets, 1794 bytes):
Chain output (policy ACCEPT: 188978 packets, 66087385 bytes):
```

Tabla 4-9 Mensaje que muestra si ipchains esta bloqueando los paquetes

En este caso se puede observar que ipchains está aceptando la entrada y salida de los paquetes de datos, por lo que no hay ningún conflicto entre dhcpd e ipchains.

- Iptables: Si la versión del kernel es mayor o igual a la 2.4, se tiene la posibilidad de manipular la entrada y salida de paquetes con ipchains o iptables, pero solo uno a la vez. Generalmente se utiliza iptables debido a que tiene mejoras en el funcionamiento y en el control disponible a la hora de filtrar paquetes.

Se debe ejecutar el siguiente comando para determinar el comportamiento de iptables:

```
iptables -L -v
```

La salida debe ser similar a la siguiente:

```
Chain INPUT (policy ACCEPT 18148 packets, 2623K bytes)
pkts bytes target prot opt in out source destination
Chain FORWARD (policy ACCEPT 0 packets, 0 bytes)
pkts bytes target prot opt in out source destination
Chain OUTPUT (policy ACCEPT 17721 packets, 2732K bytes)
pkts bytes target prot opt in out source destination
```

Tabla 4-10 Mensaje que muestra el comportamiento de iptables

Al igual que en ipchains, si se observa que se está aceptando la entrada y salida de paquetes, no habrá ningún problema entre iptables y dhcpd.

4.3.3.4 Problemas con la terminal

Si después de verificar las configuraciones en el servidor no se logra detectar en él algún problema, es posible que el conflicto se encuentre en la terminal. Para comprobar si la terminal está enviando la petición al servidor se puede observar el archivo /var/log/messages mientras la terminal se inicia.

Se debe teclear el siguiente comando:

```
tail -f /var/log/messages
```

Este comando mostrará las últimas líneas del archivo messages conforme se vayan agregando.

```
server dhcpd: DHCPDISCOVER from 00:50:56:81:00:01 via eth0
server dhcpd: no free leases on subnet WORKSTATIONS
server dhcpd: DHCPDISCOVER from 00:50:56:81:00:01 via eth0
server dhcpd: no free leases on subnet WORKSTATIONS
```

Tabla 4-11 Mensaje para saber si la terminal esta enviando la petición al servidor

Si se muestran algunas líneas en donde aparezca el mensaje "no free leases", entonces dhcpd está corriendo, pero en cambio hay un problema con la terminal debido a que no está haciendo la petición al servidor DHCP.

Esto puede deberse a que la imagen de arranque está defectuosa o la versión no es la correcta; aunque efectivamente dicha imagen corresponda a la tarjeta de red de la terminal, en algunas ocasiones la configuración con la que fue creada provoca que esta no funcione adecuadamente a la hora de arrancar la terminal.

Para solucionar esto, se debe probar iniciando la terminal con distintas versiones de la imagen de arranque; estas versiones se pueden conseguir también en la misma página[15].

En un caso extremo, si ninguna de las imágenes de arranque funciona, se puede probar reemplazando la tarjeta de red de la terminal; ya que puede tener algún problema físico que impida la conectividad con el servidor.

4.3.4 Problemas con el servicio TFTP

La imagen de arranque que permite iniciar las terminales contiene instrucciones o código Etherboot, el cual después de lograr comunicarse con el servidor utiliza el

[15] Página del sitio para descargar la imagen de arranque http://www.rom-o-matic.net.

protocolo TFTP para obtener un kernel Linux a través de la red. Este protocolo es muy simple, pero en ocasiones se generan ciertos problemas cuando se trata de hacerlo funcionar.

En el momento en que la terminal intenta bajar el kernel del servidor, se observa una línea similar a la siguiente:

```
Loading 192.168.0.254:/lts/vmlinuz.tulip .....
```

Tabla 4-12 Mensaje que muestra cuando la terminal intenta bajar el kernel

Posteriormente comienzan a desplegarse una serie de puntos que representan el avance de la descarga; mientras estos puntos continúen apareciendo en la pantalla significa que la imagen del kernel está siendo descargada exitosamente y por lo tanto TFTP está funcionando como se espera.

En el caso de que estos puntos no aparezcan de manera continua entonces quiere decir que hay un problema con el servicio TFTP, las posibles causas pueden ser las siguientes.

4.3.4.1 El demonio tftpd

En Fedora Core, tftp es iniciado por el demonio xinetd; existe un script de inicio llamado /etc/xinetd.d/tftp que contiene la información necesaria para que corra tftpd.

Es posible que el demonio tftpd no esté corriendo, por lo que se puede probar reiniciando el demonio xinetd, el cual reiniciará automáticamente un conjunto de servicios incluyendo tftp. Se debe teclear el siguiente comando:

```
service xinetd restart
```

4.3.4.2 La imagen del kernel no está en el directorio adecuado

La imagen del kernel que será descargado por la terminal, necesita estar en un lugar accesible para el demonio tftpd. Si la opción "-s" está especificada cuando tftpd inicia, entonces cualquier petición de la terminal debe ser relativa al directorio /tftpboot.

Por ejemplo, si el parámetro filename en el archivo de configuración dhcpd.conf es /lts/vmlinuz-2.4.19-ltsp-1, entonces la imagen del kernel deberá estar en la siguiente ruta: /tftpboot/lts/vmlinuz-2.4.19-ltsp-1.

Si no se especifica la opción "-s", entonces en el archivo dhcpd.conf tendrá que especificarse la ruta absoluta de la imagen del kernel, es decir:

```
filename        "/tftpboot/lts/vmlinuz-2.4.19-ltsp-1";
```

Tabla 4-13 Mensaje donde se especifica la ruta absoluta de la imagen del kernel

4.3.5 Problemas con el servicio NFS

NFS permite montar un sistema de archivos remoto a cada una de las terminales, éste sistema de archivos será realmente un directorio compartido que será exportado por el servidor; por lo que éste se debe configurar adecuadamente para que permita el acceso de las terminales.

Existen varios factores que pueden impedir que un sistema de archivos raíz pueda ser montado en las terminales:

4.3.5.1 El proceso Init no puede ser iniciado

Cuando la terminal intenta montar el sistema de archivos raíz, puede mostrar el siguiente error:

```
Kernel panic: No init found. Try passing init= option to kernel.
```

Tabla 4-14 Mensaje de error al intentar montar el sistema de archivos raíz

Este mensaje indica que el contenido del directorio exportado por el servidor no es el adecuado, o no contiene la información que necesita el kernel para continuar con el proceso de configuración del sistema.

Los scripts de configuración del LTSP pudieron haber tenido errores en la creación del directorio a exportar, por lo que es probable que el directorio /opt/ltsp/i386 esté vacío.

Se puede listar el contenido de este directorio para verificar que tenga la estructura correcta y la cual es la que espera encontrar el kernel de la terminal.

```
ls -l  /opt/ltsp/i386
```

Los siguientes directorios deberán estar presentes:

bin dev etc lib mnt oldroot opt proc root sbin tmp usr var

4.3.5.2 El servidor retorna el error -13

Si se muestra en la terminal el siguiente error:

```
Root-NFS: Server returned error -13 while mounting /opt/ltsp/i386
```

Tabla 4-15 Mensaje de error que muestra que el directorio i386 no esta en exports

Esto indicará que el directorio /opt/ltsp/i386 no está especificado en el archivo de configuración /etc/exports.

Es posible observar el contenido del archivo /var/log/messages para determinar alguna posible causa.

```
Jul 20 00:28:39 jamlap rpc.mountd: refused mount request from ws001
for /opt/ltsp/i386 (/): no export entry
```

Tabla 4-16 Muestra el contenido del archivo /var/log/messages

Este mensaje se puede interpretar fácilmente y confirma que efectivamente el directorio que la terminal pretende montar, no está siendo exportado por el servidor.

Para solucionar esto, basta con verificar el contenido del archivo /etc/exports y añadir si es que no existe, la siguiente línea:

```
/opt/ltsp/i386     192.168.0.0/255.255.255.0(ro,no_root_squash,sync)
```

Tabla 4-17 Muestra el contenido del archivo /

4.3.6 Problemas con el servidor X

La implementación del LTSP puede llegar a ser complicada debido a la gran diversidad de hardware, y un claro ejemplo de esto es la correcta configuración del servidor X.

Si se está utilizando una tarjeta de video y un monitor relativamente modernos, es casi seguro que estén soportados por el servidor XFree86 y por lo tanto no habrá mucho problema en que funcionen adecuadamente; sin embargo cuando se trata de equipos antiguos es necesario realizar algunos ajustes adicionales al servidor.

Los scripts de inicialización del LTSP tratarán de ejecutar el servidor X en las terminales, para que éste se comunique con el Display Manager del servidor. Si por alguna razón el servidor X no puede ser iniciado al primer intento, se probará hacerlo en 10 ocasiones más hasta que finalmente se muestre un mensaje de error si no se tuvo éxito en la conexión.

Cuando sucede esto, lo primero que se debe intentar es cambiar el servidor X especificado en el archivo lts.conf; por default se tendrá configurado el servidor XFree86 pero si la tarjeta de video es algo antigua es muy probable que se deba usar otro servidor X, por ejemplo Xvesa.

Antes de poder modificar el archivo lts.conf, será necesario instalar el paquete correspondiente al servidor Xvesa; el cual puede obtenerse en la página oficial del LTSP.

Para instalar el paquete se tiene que teclear el siguiente comando:

```
rpm –ivh  ltsp_x_vesa-3.0-1.k12ltsp.2.3.0.i386.rpm
```

Este paquete instalará los componentes necesarios para que el servidor Xvesa esté disponible para las terminales, para ello se creará el archivo Xvesa en la ruta:

```
/opt/ltsp/i386/usr/X11R6/bin/
```

Tabla 4-18 Muestra la ruta del archivo Xvesa

Una vez instalado el paquete, se requiere modificar el parámetro XSERVER en la sección de la terminal con problemas; se tendrá que cambiar el valor Xfree86 por Xvesa.

```
XSERVER = Xvesa
```

Después de efectuar los cambios correspondientes, se debe reiniciar la terminal para verificar si el servidor Xvesa soporta su tarjeta de video y monitor.

En caso de que el servidor Xvesa tampoco funcione, se tendrá que recurrir a otras alternativas, por ejemplo los servidores vga o svga; por lo cual es necesario repetir el proceso que se utilizó con Xvesa.

Hay ocasiones en que el servidor X si soporta a la tarjeta de video de la terminal, sin embargo la configuración del monitor no es la apropiada.

En modelos de monitores recientes, la configuración automática es suficiente, pero en monitores antiguos es necesario especificar algunos parámetros adicionales; los cuales definen la resolución así como la frecuencia de refresco de la pantalla.

Se tendrá que agregar los siguientes parámetros al archivo lts.conf, en la sección de la terminal:

```
X_HORZSYNC          = "30.9-49.0"
X_VERTREFRESH  = "50.0-100.0"
X_MODE_0          = 800x600
```

O bien se podrá especificar un modeline en el parámetro X_MODE_0, por ejemplo:

```
X_HORZSYNC  = "30.9-49.0"
X_VERTREFRESH = "50.0-100.0"
X_MODE_0 = 800x600  40     800  840  968 1056   600 601 605 628
    +hsync +vsync
```

Con las modificaciones antes descritas es muy probable que el servidor X inicie correctamente en cada una de las terminales, pero aun es necesario que éste logre establecer una conexión con el Display Manager que corre en el servidor.

4.3.7 Problemas con el Display Manager

El Display Manager (Administrador de Sesiones Gráficas) es un demonio que corre en el servidor, el cual espera que un servidor X se ponga en contacto con él. En el momento en que se establece la conexión se mostrará una caja de diálogo de inicio de sesión, para que el usuario pueda realmente acceder a las aplicaciones del servidor.

Los tres Display Managers más comunes en Linux son:

XDM - Forma parte del sistema estándar de X Window.
GDM - El "Gnome Display Manager" forma parte del entorno de escritorio Gnome.
KDM - El "KDE Display Manager" forma parte del entorno de escritorio KDE.

En ocasiones el servidor X logra iniciar correctamente en la terminal, pero no puede establecer una conexión con el Display Manager; cuando sucede esto, se presenta una pantalla gris con un cursor en forma de X en lugar de la caja de diálogo de inicio de sesión. Las posibles razones son las siguientes:

4.3.7.1 El Display Manager no está en ejecución

Para determinar si algún Display Manager está corriendo en el servidor se puede utilizar el siguiente comando:

```
netstat -ap | grep xdmcp
```

La salida de éste comando mostrará si hay algún proceso escuchando en el puerto xdmcp (177).

```
udp 0 0 *:xdmcp *:* 1255/gdm
```

Tabla 4-19 Muestra la salida para determinar si algún display manager esta activo

En este ejemplo se puede observar que gdm está corriendo con el PID 1255 y está escuchando en el puerto de XDMCP. En caso de que no se muestre salida alguna, significará que ningún Display Manager está corriendo; por lo cual será necesario reiniciar el sistema para reactivar el Display Manager por defecto que utiliza el servidor.

4.3.7.2 El Display Manager ignora peticiones de máquinas remotas

Si se ha comprobado que algún Display Manager está corriendo, entonces es posible que haya sido configurado para ignorar peticiones XDMCP desde máquinas remotas; para solucionar esto, es necesario verificar los archivos de configuración correspondientes al Display Manager utilizado.

4.3.7.2.1 XDM

En el archivo xdm-config la siguiente línea deberá estar comentada para que se admitan peticiones remotas:

```
#    DisplayManager.requestPort: 0
```

Y el archivo Xaccess deberá contener una línea que comience con un asterisco (*):

```
*    # Aceptar peticiones remotas
```

Ambos archivos generalmente se encuentran en la ruta /etc/X11/xdm/ pero si no es así, se pueden buscar con el comando locate.

4.3.7.2.2 KDM

El archivo kdmrc contiene una sección con la etiqueta [xdmcp], dentro de la cual hay un parámetro que deberá estar configurado de la siguiente manera:

```
Enable = true    #true habilita las peticiones remotas
```

Este archivo generalmente se encuentra en la ruta /etc/kde/kdm/ pero si no es así, se puede buscar con el comando locate.

4.3.7.2.3 GDM

El archivo gdm.conf contiene una sección con la etiqueta [xdmcp], dentro de la cual hay un parámetro que deberá estar configurado de la siguiente manera:

```
Enable = true    #true habilita las peticiones remotas
```

Este archivo generalmente se encuentra en la ruta /etc/X11/gdm/ pero si no es así, se puede buscar con el comando locate.

4.3.7.3 Conflicto del Display Manager con el DNS

Si el Display Manager definitivamente está en ejecución y está escuchando las peticiones de máquinas remotas, entonces el problema puede deberse a un simple error en el mapeo de direcciones IP a nombres, por lo que el Display Manager no sabrá con que terminal establecer la comunicación.

Para solucionar este problema es necesario configurar el archivo hosts, agregando una entrada para cada una de las terminales, por ejemplo:

```
192.168.0.1   ws001    ws001.localdomain   ws001
192.168.0.2   ws002    ws002.localdomain   ws002
```

4.3.8 Archivos de sistema

Finalmente, después de resolver los problemas detectados y realizar las modificaciones correspondientes a los archivos de configuración estos quedarán así:

```
# Archivo de configuración dhcpd.conf
ddns-update-style            none;
default-lease-time           21600;
max-lease-time               21600;
option subnet-mask           255.255.255.0;
option broadcast-address     192.168.0.255;
option routers               192.168.0.254;
option domain-name-servers   192.168.0.254;
option domain-name           "yourdomain.com";
option root-path             "192.168.0.254:/opt/ltsp/i386";
option option-128 code 128 = string;
option option-129 code 129 = text;

shared-network WORKSTATIONS {
   subnet 192.168.0.0 netmask 255.255.255.0 {
   }
}

group    {
   use-host-decl-names      on;
   option log-servers       192.168.0.254;
   host ws001 {
      hardware ethernet     00:60:8C:CA:BC:FF;
      fixed-address         192.168.0.1;
      filename              "/lts/vmlinuz-2.4.19-ltsp-1";
      option option-128     e4:45:74:68:00:00;
      option option-129     "NIC=3c509";
   }
   host ws002 {
      hardware ethernet     00:A0:24:24:F6:0F;
      fixed-address         192.168.0.2;
      filename              "/lts/vmlinuz-2.4.19-ltsp-1";
      option option-128     e4:45:74:68:00:00;
      option option-129     "NIC=3c509";
   }
   host ws003 {
      hardware ethernet     00:80:AD:3B:8F:E4;
      fixed-address         192.168.0.3;
      filename              "/lts/vmlinuz-2.4.19-ltsp-1";
   }
}
```

Tabla 4-20 Muestra le archivo completo de configuración exports

```
# Archivo de configuración exports

/opt/ltsp/i386                  192.168.0.0/255.255.255.0(ro,no_root_squash,sync)
/var/opt/ltsp/swapfiles   192.168.0.0/255.255.255.0(rw,no_root_squash,async)
```

Tabla 4-21 Muestra el archivo completo de configuración exports

```
# Archivo de configuración hosts.allow

bootpd:   0.0.0.0
in.tftpd:  192.168.0.
portmap:   192.168.0.
mountd:    192.168.0.
ALL:        127.0.0.1
ALL:      192.168.0.254
ALL:      192.168.0.1
ALL:      192.168.0.2
ALL:      192.168.0.3
```

Tabla 4-22 Muestra el archivo completo de configuración hosts.sllow

```
# Archivo de configuración lts.conf

[Default]
      SERVER          = 192.168.0.254
      XSERVER         = auto
      X_MOUSE_PROTOCOL  = "PS/2"
      X_MOUSE_DEVICE    = "/dev/psaux"
      X_MOUSE_BUTTONS   = 3
      X_USBMOUSE_PROTOCOL= "IMPS/2"
      X_USBMOUSE_DEVICE  = "/dev/input/mice"
      X_USBMOUSE_BUTTONS = 5

      XkbSymboles      = "us(pc101)"
      XkbModel         = "pc101"
      XkbLayout        = "us"

      USE_XFS          = N
      LOCAL_APPS       = N
      RUNLEVEL         = 2

      SOUND            = Y

      VOLUME           = 75

[ws001]
      XSERVER          = Xvesa
      LOCAL_APPS       = N
      USE_NFS_SWAP     = N
      SWAPFILE_SIZE    = 48m
      RUNLEVEL         = 5
      X_HORZSYNC       = "30.9-49.0"
      X_VERTREFRESH    = "50.0-100.0"
      X_MODE_0         = 800x600
[ws002]
      XSERVER          = Xvesa
      LOCAL_APPS       = N
```

```
    USE_NFS_SWAP      = N
    SWAPFILE_SIZE     = 48m
    RUNLEVEL          = 5
    X_HORZSYNC        = "30.9-49.0"
    X_VERTREFRESH     = "50.0-100.0"
    X_MODE_0          = 800x600
[ws003]
    XSERVER           = Xvesa
    LOCAL_APPS        = N
    USE_NFS_SWAP      = N
    SWAPFILE_SIZE     = 48m
    RUNLEVEL          = 5
    X_MODE_0          = 800x600
```

Tabla 4-23 Muestra el archivo completo de configuración lts.conf

4.4 Conexión en red

De acuerdo con la metodología propuesta, cuando se logra dar solución a los problemas detectados y todas las terminales inician una sesión gráfica individualmente; el siguiente paso es probar su funcionamiento conectándolas en red.

Este proceso es relativamente sencillo, pero hay que considerar algunos aspectos importantes en su implementación, los cuales básicamente son:

- Uso de cable UTP directo en lugar de cable cruzado
- Uso de un switch en lugar de un hub para garantizar un mejor rendimiento en la red
- Asignación correcta de direcciones IP para evitar conflictos entre las terminales

Tomando en cuenta los puntos anteriores, los pasos para implementar la red son los siguientes:

1. Conectar el servidor al primer puerto del switch mediante un cable directo
2. Conectar la terminal 1 al segundo puerto del switch, la segunda terminal al tercer puerto y así sucesivamente, todas estas conexiones mediante cable directo
3. Encender el servidor e iniciar los servicios necesarios del LTSP
4. Encender el switch
5. Encender cada una de las terminales y autenticarse para iniciar una sesión remota en el servidor

El siguiente diagrama muestra la configuración de los equipos al estar conectados en red:

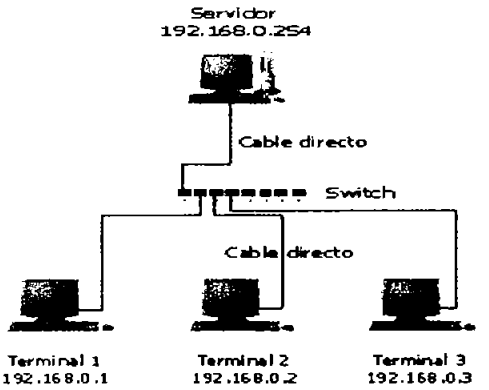


Figura 4-2 Diagrama que muestra la configuración de los equipos en red

La siguiente figura muestra a las terminales con la pantalla de bienvenida del LTSP al estar en red.

Figura 4-3 Imagen de las terminales en red con la pantalla de bienvenida del LTSP

4.5 Conexión a internet

Inicialmente, la red de terminales gráficas ha sido configurada para funcionar de manera local, esto de por si ya ofrece muchas ventajas, las cuales han sido mencionadas a lo largo de este trabajo, pero debido a que actualmente el acceso a Internet es indispensable, se optó por configurar la red local para que tenga acceso a éste.

Para establecer una conexión a Internet, debe existir un medio por el cual la red local se pueda comunicar con las redes externas; esto básicamente se logra con una tarjeta de red adicional en el servidor, lo que permite que éste también tenga la función de un gateway. Aunque es posible utilizar una sola tarjeta de red creando un "alias", esto no es recomendable debido al tráfico y a la carga que se generaría en dicho dispositivo.

Al configurar el servidor para que tenga salida a Internet por medio de la segunda tarjeta de red (eth1), automáticamente todas las terminales también tendrán acceso. Es necesario entender que en una red implementada con el LTSP, cualquier usuario conectado en un ambiente grafico, es en realidad un usuario local del servidor, por lo tanto, cualquier aplicación que lo requiera puede usar la misma conexión a Internet del servidor sin ningún inconveniente.

El programa redhat-config-network permite configurar de forma sencilla los dispositivos de red instalados en el sistema, éste puede ser ejecutado desde una terminal con el siguiente comando:

```
redhat-config-network &
```

Este programa detecta de forma automática los dispositivos de red instalados y los muestra en la sección "Dispositivos", en este caso deben aparecer dos dispositivos: eth0 y eth1.

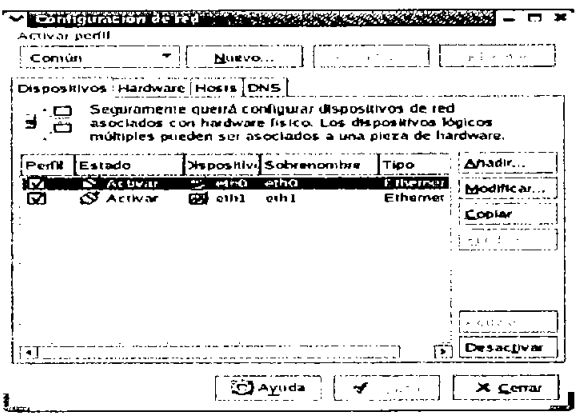


Figura 4-4Pantalla donde se muestran los dispositivos lógicos de red

En la pestaña Hardware, se debe verificar que se haya detectado correctamente la marca y modelo de las tarjetas de red; esto es necesario debido a que pueden ocurrir errores en los controladores, si estos no son los apropiados.

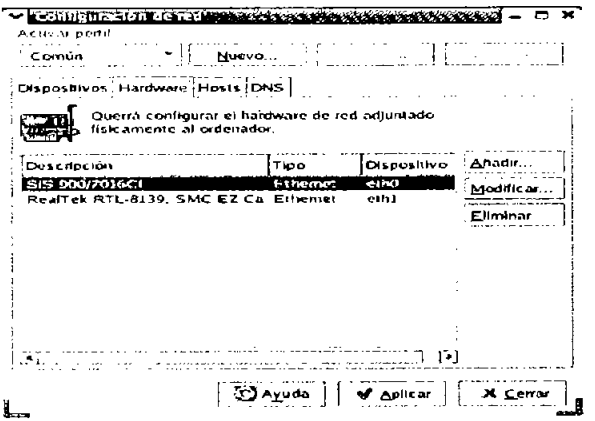


Figura 4-5 Pantalla donde se muestran los dispositivos físicos de red

La tarjeta eth0 permitirá comunicar al servidor con la red de terminales gráficas, para configurarla se debe regresar a la pestaña "Dispositivos" y seleccionar el dispositivo eth0; hacer click en "Modificar".

En la siguiente pantalla, se tiene que desactivar la opción "Obtener las configuraciones de direcciones IP automáticamente con dhcp", se activará la opción "Configurar las direcciones IP de manera estática". Se debe ingresar la dirección IP 192.168.0.254, la máscara de subred 255.255.255.0 y la puerta de enlace 192.168.0.254. Hacer click en "OK" para continuar.

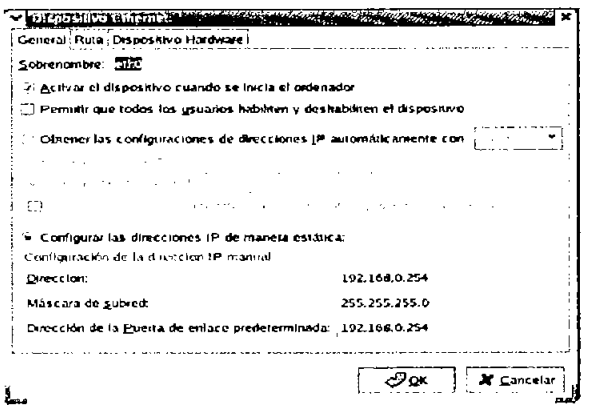


Figura 4-6 Configuración del dispositivo eth0

La tarjeta eth1 permite comunicar al servidor con las redes externas, en realidad es en este dispositivo donde se configura el acceso a Internet, por lo que se debe seleccionar y hacer click en "Modificar". En la siguiente pantalla, se tiene que desactivar la opción "Obtener las configuraciones de direcciones IP automáticamente con dhcp", se activará la opción "Configurar las direcciones IP de manera estática". Se debe ingresar la dirección IP 132.248.173.84, la máscara de subred 255.255.255.0 y la puerta de enlace 132.248.173.254. Hacer click en "OK" para continuar.

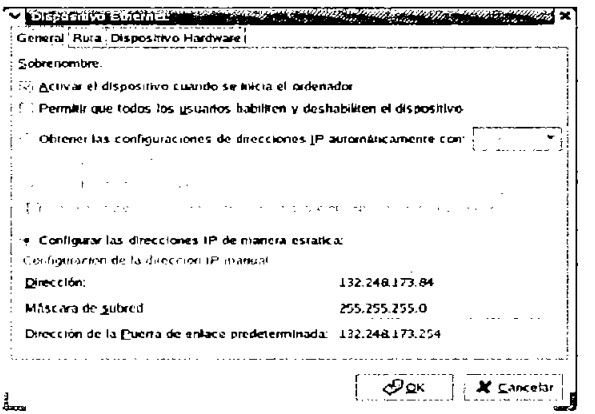


Figura 4-7 Configuración del dispositivo eth1

En la pestaña "DNS", se debe ingresar el nombre de host "Electromartin", el DNS primario 132.248.204.1 y el DNS secundario 132.248.10.2. En la sección "Ruta de búsqueda DNS", se debe añadir el dominio "aragon.unam.mx". Una vez ingresados los datos, hacer click en "Aplicar" para confirmar los cambios.

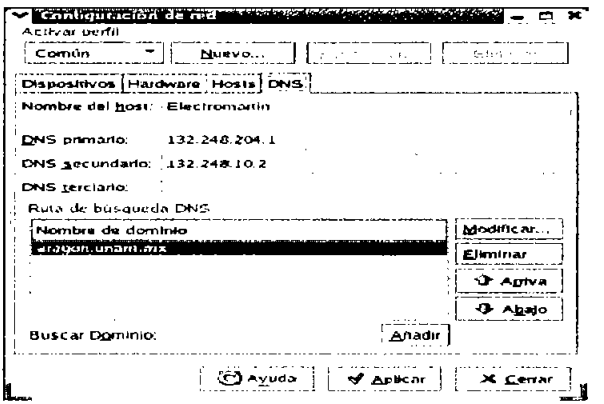


Figura 4-8 Configuración del DNS

4.6 Software específico

El LTSP es un claro ejemplo de las ventajas y posibilidades que el software libre puede ofrecer, ya que mediante la adecuada combinación de diferentes servicios y programas se logró implementar la red de terminales gráficas en el laboratorio de electrónica.

Como ya se había mencionado, dicha red proporciona entre otras cosas, un sistema centralizado de almacenamiento de información y la posibilidad de utilizar una gran variedad de aplicaciones, la mayoría de ellas distribuidas bajo la licencia GNU/GPL.

Sin embargo, no es conveniente instalar de una forma indiscriminada programas que tal vez nunca serán utilizados; para evitar esto se pensó en la necesidad de los alumnos de contar con software relacionado con la electrónica, el cual puede clasificarse en tres principales grupos: diseño, simulación y programación de circuitos.

Actualmente ya se cuenta con software de este tipo, por ejemplo: el Electronics Workbench, Pspice, Warp, Mplab, entre otros; pero debido a que estos programas corren en Windows se optó por buscar software propio de Linux que pudiera ser útil en el laboratorio y que tuviera características semejantes a los anteriores.

Cabe mencionar que el software para Windows podrá seguir utilizándose debido a que las terminales han sido configuradas para que puedan seguir arrancando este sistema operativo si así se requiere. En Internet hay muchos programas de electrónica disponibles, pero solo se describirán los que se consideraron más funcionales y amigables para el usuario.

4.6.1 Diseño

Este tipo de software es utilizado para crear diagramas o esquemas que representan gráficamente un circuito electrónico, éstos pueden imprimirse o exportarse a otros programas mediante la generación de un archivo compatible. Los programas de diseño no tienen la capacidad de simular o interpretar el funcionamiento del diagrama, sin embargo son muy útiles debido a que permiten generar fácilmente la estructura de casi cualquier circuito eléctrico - electrónico.

4.6.1.1 Eagle

Es un software de diseño de esquemas PCB (Printed Circuit Boards) con motor de autotrazado y una gran variedad de componentes. Es comercial, pero existe una licencia freeware (educacional) que limita el tamaño máximo de la placa de circuito impreso.

El programa consta de tres módulos principales:

- Editor de circuito impreso (Layout Editor)
- Editor de diagramas esquemáticos (Schematic Editor)
- Organizador de pistas (Autorouter)

Estos tres módulos están integrados en una sola interfaz de usuario, lo que permite crear fácilmente el diseño para la placa, a partir del circuito implementado en el editor esquemático.

La versión de evaluación del software permite crear circuitos impresos de hasta dos caras y con área máxima de 100 x 80 mm, suficiente para una gran variedad de circuitos sencillos.

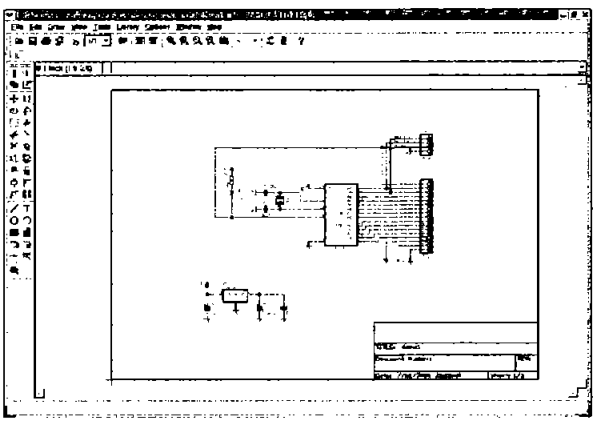


Figura 4-11 Imagen del programa Eagle

Para la instalación de Eagle, se debe descargar la última versión del código fuente[16] y seguir los pasos descritos en el capítulo 2 (métodos de instalación).

4.6.1.2 Xcircuit

Xcircuit es un programa que permite dibujar de una manera sencilla, diagramas de circuitos eléctricos de gran calidad, los cuales pueden ser exportados a otros programas como el Spice. También es posible realizar esquemas de diversas temáticas, debido a que pueden utilizarse bibliotecas de distintos objetos.

[16] Página para descargar el código http://www.cadsoft.de/info.htm

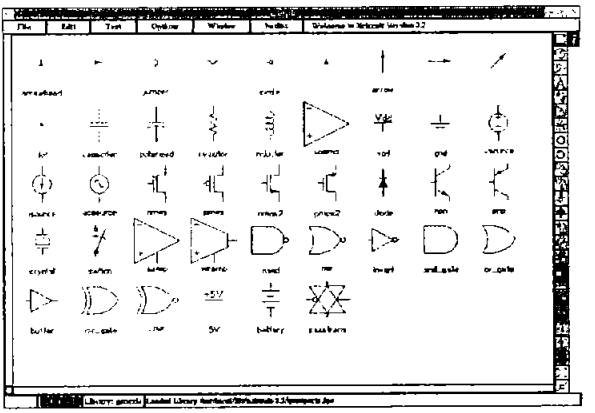


Figura 4-12 Imagen del programa Xcircuit

Para la instalación de Xcircuit, se debe descargar la última versión del código fuente[17] y seguir los pasos descritos en el capítulo 2 (métodos de instalación).

4.6.2 Simulación

Estos programas también permiten crear diagramas de circuitos electrónicos, pero tienen la gran ventaja de que se puede analizar el comportamiento físico del circuito; se puede determinar el voltaje, corriente o los estados lógicos por los que va pasando cada componente. Por lo general, los simuladores cuentan con varias herramientas entre las cuales se encuentran: los diagramas de componentes, diagramas de onda y cálculo de parámetros.

4.6.2.1 Klogic

Klogic es una herramienta para construir y simular circuitos digitales que contienen componentes estándar como AND, OR, XOR, Flip Flops RS y JK. Si se requiere construir circuitos más complejos y reutilizables se tiene la opción de diseñar subcircuitos.

Mientras se está construyendo el circuito, la simulación está ejecutándose de forma permanente, además para pruebas extendidas el programa tiene la capacidad de simular el circuito paso a paso para reconocer los estados por los que éste va pasando. También es posible desplegar la señal de flujo de los componentes de un circuito de forma gráfica e intuitiva.

[17] Página para descargar el código http://bach.ece.jhu.edu/~tim/programs/xcircuit/welcome.html

paquetes; lo que permite que su uso sea fácilmente extendido para nuevas áreas de aplicación.

Los principales componentes de Ksimus son:

- Compuertas AND, NAND, OR, NOR, XOR, XNOR
- Flip Flops RS, JK y JK-MS
- Generador de pulsos
- LED
- Latches D
- Contador
- Display de 7 segmentos
- Multiplexor y demultiplexor
- Generador de ondas
- Aritméticos (add, sub, mul, div, mod, abs)
- Trigonométricos (sin, asin, cos, acos, tan, atan, atan2)
- Hiperbólicos (sinh, asinh, cosh, acosh, tanh, atanh)
- Exponenciales (log, log10, exp, pow, square)

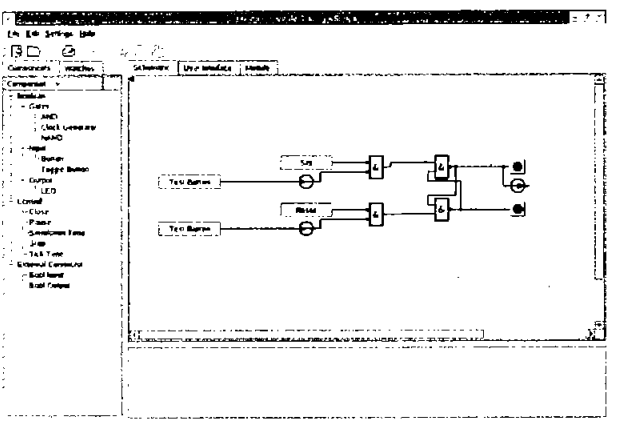


Figura 4-15 Imagen del programa Ksimus

Para la instalación de Ksimus, se debe descargar la última versión del código fuente[19] y seguir los pasos descritos en el capítulo 2 (métodos de instalación).

[19] Página para bajar el código fuente http://ksimus.berlios.de/index.html

4.6.2.3 Tkgate

Tkgate es un simulador de circuitos digitales, basado en un editor gráfico tcl/tk. Soporta una amplia gama de componentes básicos, pero además se tiene la capacidad de construir módulos mediante el diseño jerárquico de circuitos.

Los principales componentes de Tkgate son:

- Compuertas AND, NAND, OR, NOR, XOR, XNOR
- Transistores NMOS y PMOS
- Buffers triestado
- Componentes ALU (sumadores y multiplicadores)
- Elementos de memoria (registros, RAMs, ROMs)

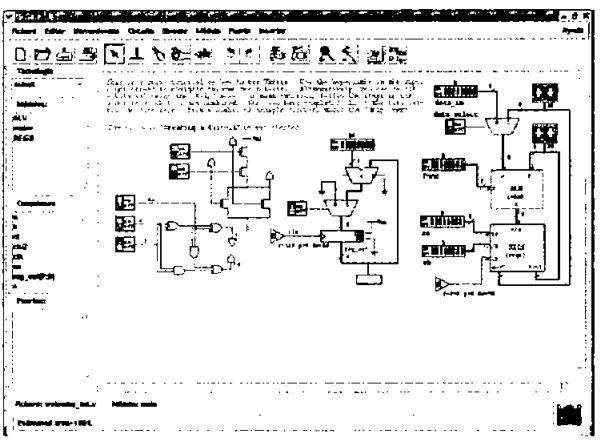


Figura 4-16 Imagen del programa Tkgate

Para la instalación de Tkgate, se debe descargar la última versión del código fuente[20] y seguir los pasos descritos en el capítulo 2 (métodos de instalación).

4.6.2.4 Vipec

Vipec es una poderosa herramienta para el análisis de circuitos de alta frecuencia y redes eléctricas lineales. Es adecuado para simular en el dominio de la frecuencia, circuitos relacionados con las comunicaciones, como: amplificadores, filtros y líneas de transmisión.

[20] Página para bajar el código fuente http://www.tkgate.org

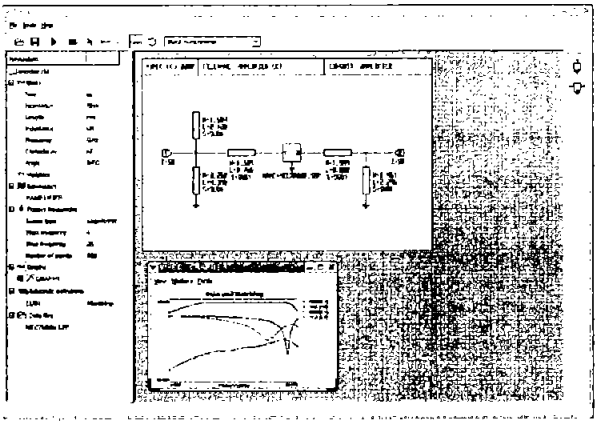


Figura 4-17 Imagen del programa Vipec

Para la instalación de Vipec, se debe descargar la última versión del código fuente[21] y seguir los pasos descritos en el capítulo 2 (métodos de instalación).

4.6.2.5 Gpsim

Este entorno de simulación para PIC's soporta microprocesadores de 12, 14 y 16 bits, además tiene la capacidad de simular en tiempo real sistemas con un ciclo de reloj de 20 MHz.

GPSIM cuenta con un depurador, un desensamblador, un manejador de estímulos, un mapa de memoria y algunos plugins que permiten simular componentes externos al PIC como una pantalla LCD.

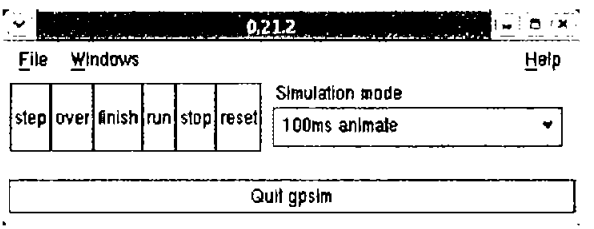


Figura 4-18 Imagen del programa Gpsim

[21] Página para bajar el código fuente http://vipec.sourceforge.net

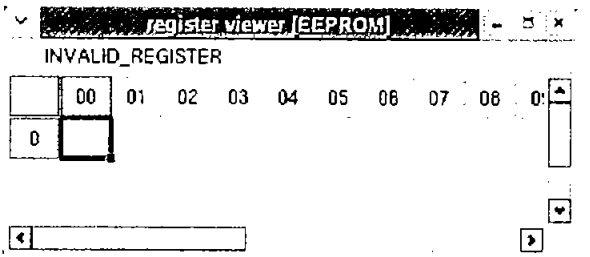


Figura 4-19 Imagen de un ejemplo de programación del programa Gpsim

Para la instalación de Gpsim, se debe descargar la última versión del código fuente[22] y seguir los pasos descritos en el capítulo 2 (métodos de instalación).

4.6.3 Programación

Estas aplicaciones permiten diseñar mediante lenguajes de alto nivel, la estructura lógica de circuitos programables, por ejemplo: memorias, microcontroladores, microprocesadores, PAL, GAL, etc.

Una vez que el diseño es terminado, éste tiene que ser copiado en el chip correspondiente, lo cual es posible utilizando un dispositivo especial denominado "grabador". En ocasiones es posible simular el comportamiento del circuito antes de ser grabado, lo que permite la optimización y depuración del diseño.

4.6.3.1 BlueHDL

BlueHDL es una suite de herramientas VHDL, que consiste en un editor y compilador VHDL, un motor de simulación, y una interfaz de visualización de ondas.

Permite diseñar con lenguajes de alto nivel logrando desarrollar circuitos programables, es una buena opción ya que para todo circuito por sencillo que sea se busca realizarlo programando un microcontrolador.

[22] Página para bajar el código fuente http://www.dattalo.com/gnupic/gpsim.html

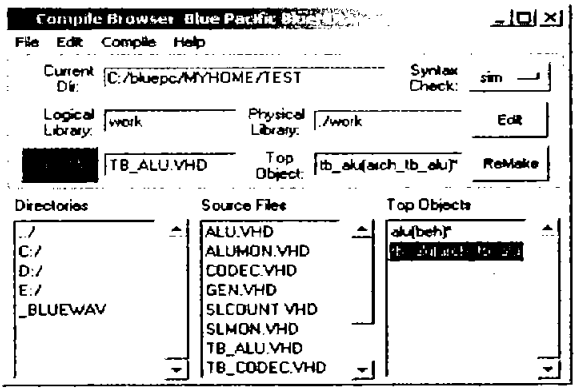


Figura 4-20 Imagen del programa BlueHDL

Para la instalación de BlueHDL, se debe descargar la última versión del código fuente[23] y seguir los pasos descritos en el capítulo 2 (métodos de instalación).

Con todas las instalaciones y configuraciones previas se comprobó que efectivamente cada una de las terminales inició una sesión gráfica adecuadamente, apoyándonos en la parte de detección y solución de problemas; además la red y la conexión a internet funcionó tal y como se esperaba. Ahora analizaremos el rendimiento del servidor para conocer cuántas terminales realmente pueden funcionar adecuadamente.

[23] Página para bajar el código fuente http://www.bluepc.com/index.html

PRUEBAS Y ANÁLISIS DE LOS RESULTADOS

OBJETIVO ESPECÍFICO

En primer lugar se comprobará el rendimiento del servidor con sus tres terminales ejecutando diversas aplicaciones, también se analizarán algunas medidas y acciones que garanticen un mínimo de seguridad y funcionalidad en la red; y por último se darán las pautas para retomar posibles proyectos relacionados o simplemente para mejorar los servicios que proporciona el servidor.

5

INTRODUCCIÓN

Para el desarrollo de este apartado se planteó la necesidad de analizar el rendimiento del servidor para tres terminales, la idea básica es dar a conocer el porcentaje de recursos que requieren las terminales para funcionar apropiadamente sin colapsarse o alentar al servidor. Los valores se obtuvieron con la ayuda del comando TOP y en el caso del procesador sus valores son dados en porcentajes para una mejor comprensión.

Por otro lado se puntualizan las ventajas de utilizar el sistema operativo Linux que proporciona algunas características con respecto a la administración y seguridad del sistema, no es necesario realizar configuraciones complejas para garantizar la seguridad que se desea, ya que el propio sistema viene con valores predefinidos que proporcionan un nivel aceptable de seguridad; pero no está de más tomar ciertas medidas y precauciones las cuales se darán en dicho apartado.

Con respecto a la administración, Linux es un sistema realmente amigable y potente, facilita el manejo de usuarios y lo más importante es que mantiene espacios restringidos para un mejor control del sistema. Es importante mencionar que la administración y la seguridad están íntimamente ligados ya que sino hubiera un control estricto del espacio de usuarios, ésto podría ser una causa para que se realizara un ataque al sistema comprometiendo la seguridad del mismo.

Una vez que se ha comprobado que el sistema es realmente estable y se tiene un control de los archivos de sistema, así como de los recursos que se le brindan a los usuarios para no permitirles un uso inapropiado de los mismos; se puede garantizar que se cuenta con una red estable, fácil de administrar y con un nivel aceptable de seguridad.

Pero eso no es todo, ya que es posible hacer mejoras o innovaciones; hay varias aplicaciones u opciones que se pueden implementar para mejorar los servicios brindados en la red de terminales gráficas.

5.1 Pruebas de rendimiento

En este apartado se probará el rendimiento del servidor y para esto se tendrá que encender terminal por terminal para tener la oportunidad de ir monitoreando las peticiones de recursos que se hacen al servidor, y a su vez tener una relación de la cantidad de memoria que se requiere para soportar n cantidad de terminales funcionando adecuadamente.

Estas pruebas ayudarán a comprobar que Linux administra eficientemente los recursos disponibles, principalmente de memoria y de procesamiento del sistema, los valores de las tablas muestran claramente cuánto se consume de los recursos del servidor pero a la vez es posible visualizar que se llega a un estado de estabilidad que impide que el sistema deje de funcionar.

El servidor cuenta con las siguientes características:

Procesador	700 Mhz
Memoria	376640 Kb
Swap	786432 Kb

Se recurrió al siguiente comando para obtener la información del uso de procesador, memoria y swap del servidor:

```
top
```

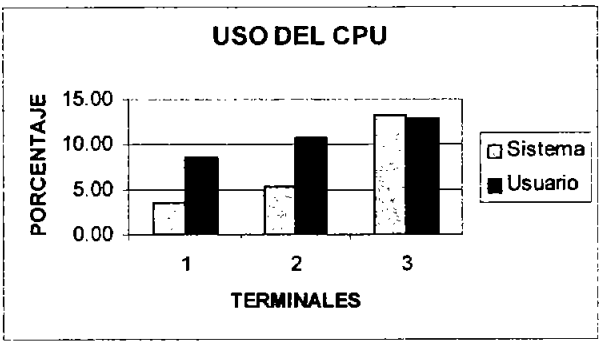
Los resultados obtenidos al momento de encender las tres terminales, se muestran en la siguiente tabla:

TERMINAL	CPU (%)		MEM (Kb)	SWAP (Kb)
	SISTEMA	USUARIO		
1	3.50	8.50	348868	0
2	5.30	10.80	364496	0
3	13.30	12.90	371024	0

Tabla 5-1 Resultados obtenidos al momento de encender las tres terminales

En la siguiente gráfica se puede notar que el procesador no tiene gran desgaste al arrancar las terminales y esto sin tomar en cuenta lo que requiere el servidor. Se puede ver que con la primer terminal se utilizó un 8.5% entonces con una segunda debería de consumir un 17% y a su vez con una tercera gastaría un 25.5% de la capacidad del procesador.

Pero esto no es así ya que realmente sólo se empleo un 13% aproximadamente y no el 25% que se tenía estimado, este comportamiento se debe principalmente a que el kernel de Linux administra eficientemente el uso del procesador; el planificador de procesos es el encargado de gestionar el uso de este recurso ya que verifica la prioridad de cada uno de los procesos y los manda a ejecución solo cuando es necesario, optimizando en gran medida los tiempos del procesador.



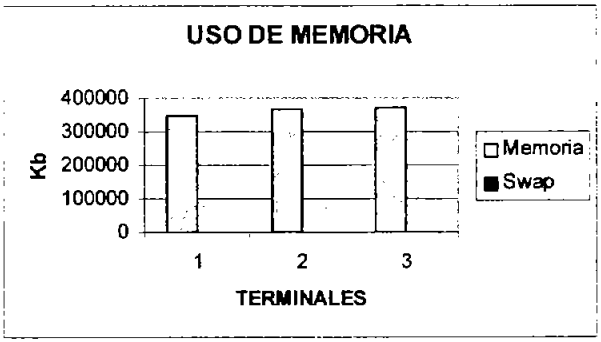
Gráfica 5-1 El uso del CPU

Se puede observar también que la memoria se está utilizando prácticamente en su totalidad, a pesar de que solo es el arranque y aún no se ha ejecutado alguna aplicación en las terminales; cabe mencionar que el sistema aun no ha requerido el uso de la memoria swap.

Bajo estas circunstancias se podría pensar que el sistema esta desperdiciando la memoria, pero no es así, ya que Linux utiliza toda la memoria libre para caché de una forma más dinámica. En un estado estable prácticamente toda la memoria principal está ocupada con fragmentos de procesos, por lo que se tendrá acceso directo a la mayor cantidad de procesos posibles disminuyendo el tiempo en la ejecución de un programa.

El kernel de Linux soporta ejecutables cargados por demanda de páginas, donde solo aquellos segmentos de un programa que se requieren son leídos del disco y puestos en la memoria.

Así, si varias instancias de un programa son ejecutadas simultáneamente, sólo una copia del código del programa estará en memoria y por esta razón el sistema se mantiene estable al ejecutar los mismos programas en las terminales.



Gráfica 5-2 El uso de la memoria

Una vez que las terminales han iniciado una sesión gráfica, es necesario verificar el rendimiento del servidor pero en este caso corriendo aplicaciones del usuario. Para esto, se eligieron 3 aplicaciones que se consideraron como las más utilizadas por los alumnos (Navegador Web, procesador de textos y hoja de cálculo), así como 3 programas específicos para el diseño y simulación de circuitos electrónicos (Xcircuit, Tkgate y Bluehdl).

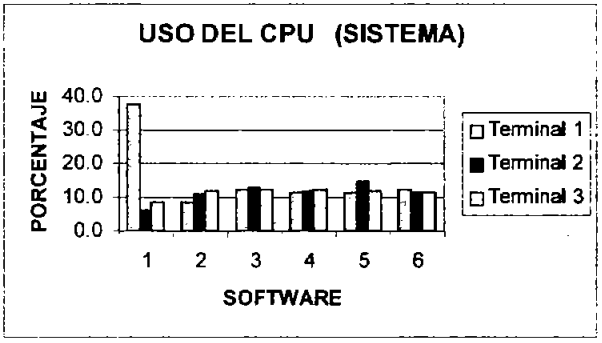
La prueba consistió en ejecutar una aplicación en la primer terminal, en este caso el navegador, luego se anotaron los valores de los recursos actuales (procesador, memoria y swap); posteriormente se hizo lo mismo con la segunda y tercer terminal. Después que se hicieron las mediciones correspondientes al primer programa, se prosiguió a ejecutar el procesador de texto, continuando con la misma mecánica se ejecutaron las aplicaciones restantes; hasta que al final se recopiló la información de cuanto fue el consumo de recursos por las tres terminales y a su vez para las 6 aplicaciones.

La siguiente tabla muestra los datos arrojados en la prueba antes descrita:

SOFTWARE	CPU (%)						MEM (Kb)			SWAP (Kb)		
	Term 1		Term 2		Term 3		Term 1	Term 2	Term 3	Term 1	Term 2	Term 3
	SIST	USR	SIST	USR	SIST	USR						
Navegador	37.6	20.5	6.0	15.9	8.7	19.7	371040	371120	365912	0	0	0
Proc. de texto	8.7	19.7	11.0	12.2	12.1	13.6	372036	371980	372216	0	0	0
Hoja de cálc.	12.2	15.2	12.9	11.6	12.5	14.7	372200	371180	371464	0	0	1132
Xcircuit	11.5	15.0	12.1	13.6	12.5	19.1	372152	372092	370964	1132	1132	1104
Tkgate	11.4	14.5	14.6	14.8	11.7	19.6	371568	371520	371508	1104	1104	1104
Bluehdl	12.4	14.8	11.4	16.8	11.2	16.1	371412	371792	371712	1104	1104	1104

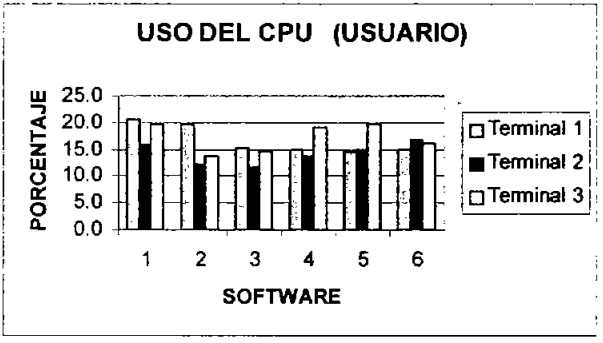
Tabla 5-2 Resultados obtenidos de las pruebas realizadas

Como puede observarse, al ejecutar el navegador en la primer terminal, se obtuvo un alto porcentaje en el uso del procesador; esto se debió a que el sistema le dio prioridad máxima a esta aplicación e incremento la actividad de procesamiento, pero después de cierto tiempo el sistema se volvió a estabilizar gracias a las características del planificador de procesos antes mencionadas.



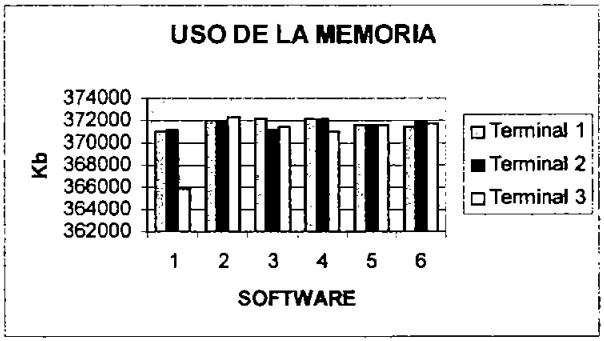
Gráfica 5-3 Uso del CPU (sistema) con las tres terminales

En el caso del uso del procesador por parte de las aplicaciones de usuario, se observa que no hubo mucha variación en el porcentaje; ya que se tiene la ventaja de que se ejecutaron las mismas aplicaciones para las tres terminales, optimizando en gran medida el uso de este dispositivo. Además con estas pruebas también fue posible comprobar la eficiencia de Linux para manejar la concurrencia en las aplicaciones.



Gráfica 5-4 Uso del CPU (usuario) con las tres terminales

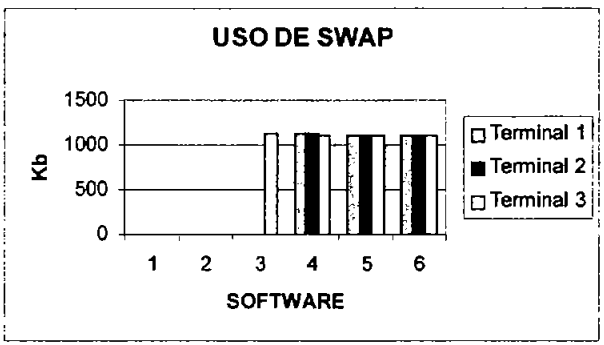
Con respecto a la memoria, se puede notar que al ejecutar cada uno de los programas ésta prácticamente se ha agotado, pero el sistema aún tiene un buen funcionamiento ya que de cierta forma llega a un punto de estabilización que impide que el sistema se colapse.



Gráfica 5-5 Uso de la memoria con las tres terminales

Sin embargo cuando la memoria física se ha terminado, Linux comienza a hacer uso de la memoria swap, sin ella las terminales se hubieran colapsado cuando la tercer terminal hiciera la petición para ejecutar el tercer programa; con esta memoria auxiliar es posible seguir trabajando y tener soporte para ejecutar tres programas más con las tres terminales funcionando adecuadamente.

La memoria swap es importante pero no puede reemplazar a una memoria física, pero puede permitir al sistema ejecutar aplicaciones más grandes guardando en disco duro aquellas partes de código que no se utilizan.



Gráfica 5-6 Uso de swap con las tres terminales

Considerando que se empezó a utilizar la memoria swap a partir del tercer programa y se cuenta con 384 MB de RAM en el servidor, se puede concluir que con un promedio de 3 aplicaciones corriendo, se necesita aproximadamente 100 MB de memoria por cada terminal; de esta manera si en un futuro se requiere expandir la red, se recomienda que por cada terminal que se incorpore, se incremente proporcionalmente el tamaño de la memoria, por ejemplo para 10 terminales se deberá tener aproximadamente 1 GB de RAM.

En cuanto al procesador, se recomienda que para más de 5 terminales, su velocidad sea mayor o igual a 1 GHz, con esto se garantiza un buen desempeño y confiabilidad de la red, solo si se desea utilizar una computadora personal como servidor.

Pero si se desea implementar este proyecto de una manera más formal, es decir, más de 15 terminales, lo más adecuado es invertir en un servidor, ya que es un equipo diseñado para estar conectado las 24 horas del día durante todo el año, el cual tiene que soportar fuertes picos de actividad, y debe tener un alto rendimiento en ejecución de aplicaciones para que los alumnos no vean afectada la velocidad y que cuente básicamente con las siguientes características, tomando como referencia las pruebas que se realizaron.

Procesador: Intel Pentium IV
Frecuencia: 3 Ghz
Memoria: 4GB
Disco Duro: 160 GB

La empresa Hewlett Packard ofrece un servidor económico de $17 000, con estas características realmente no es mucha la inversión tomando en cuenta que de esta forma se podrá tener varias 486 funcionando como estaciones de trabajo, si se invirtiera esta cantidad en computadoras medianamente potentes solo se lograría adquirir aproximadamente 4, sin tomar en cuenta la cantidad faltante por costos de licencias. Qué mejor inversión que la de adquirir este servidor y poner a funcionar esas viejas 486 brindando una potencia aceptable y sin gastos por licencias de software.

5.2 Seguridad

Con el fin de garantizar que la red de terminales gráficas funcione adecuadamente y que los recursos del servidor estén disponibles en cualquier momento, se pretende dar algunas ideas generales de los riesgos existentes y las medidas de seguridad para prevenir los posibles ataques al sistema.

La meta de la mayoría de los ataques a un sistema operativo es conseguir un acceso con privilegios de root, lo que garantiza un control total del sistema; primero se

intentará conseguir acceso como usuario normal e ir incrementando los niveles de privilegio utilizando las posibles vulnerabilidades del sistema.

Se puede decir que con el simple hecho de intentar violar la privacidad de la información, obteniendo privilegios que no le corresponden a un determinado usuario, haciendo un uso desmedido de los recursos o modificando información legítima contenida en una máquina, como puede ser el contenido de una página web o una base de datos; éstas son algunas formas de poner en riesgo el correcto funcionamiento del sistema.

En todo sistema la seguridad simplemente es fundamental, se debe tomar en cuenta que en un ambiente LTSP pueden haber muchos usuarios conectados al servidor al mismo tiempo; por lo que el sistema operativo del servidor tendrá la obligación de proteger a unos usuarios frente a otros y protegerse así mismo.

En primer lugar se debe tener en cuenta que ningún sistema es completamente seguro, solo lo será cuando no este conectado a la red o simplemente este apagado. Desde esta perspectiva se puede partir de que lo único que se puede hacer es aumentar la dificultad para que alguien pueda comprometer la seguridad del sistema.

Por tal motivo, es conveniente establecer algunas políticas que fijen el nivel de seguridad que se requiere, partiendo desde las políticas de administración del servidor, así como algunas políticas de uso para los alumnos, los cuales serán los que potencialmente podrán comprometer la seguridad de la red del laboratorio.

Lo que se desea es garantizar que el sistema permanezca funcionando de forma adecuada con comunicaciones seguras, donde nadie pueda obtener o modificar información a la que no tiene derecho. Una buena planeación ayuda bastante y antes de asegurar el sistema se debe determinar contra qué quiere protegerse y qué riesgos acepta.

Se pueden mencionar algunos requisitos de seguridad:

- Disponibilidad: Mantener la información y los recursos garantizando que no se limite el acceso autorizado a la información y el correcto funcionamiento de los recursos.
- Integridad: Requiere que la información sólo pueda ser modificada por las entidades autorizadas.
- Autenticidad: La información debe permanecer protegida ante falsificaciones, requiere de mecanismos de identificación asegurando que las comunicaciones se realizan entre entidades legítimas.
- Confidencialidad: Pretende evitar la difusión no autorizada de la información, requiere que ésta sea accesible únicamente por las entidades autorizadas.

Además en Linux se deben proteger ciertos archivos que contienen información de los usuarios (/etc/passwd, /etc/shadow), los archivos de configuración del sistema (los contenidos en etc.), el acceso al sistema y la correcta utilización de los recursos.

En muchas ocasiones los peores ataques al sistema los realiza el administrador, cuando por descuido, ignorancia o exceso de confianza altera el sistema de forma inapropiada.

Para evitar estos problemas solo hay que seguir ciertas normas de prevención como:

- No usar la cuenta de root como predeterminada, intentar primero cualquier acción como un usuario normal, si no tiene permiso puede hacer uso del comando "su" si es necesario.
- Ejecutar los comandos de forma segura verificando previamente la acción que se va a realizar.
- Activar la opción (-i) para actuar de forma interactiva con ciertos comandos que la admiten.

Al administrar un sistema Linux se deben tomar ciertas medidas que garanticen un mínimo de seguridad y funcionalidad, en primer lugar se puede limitar el espacio asignado a los usuarios ya que un posible ataque a cualquier sistema es intentar consumir todo el espacio del disco duro, una medida de seguridad es separar el árbol de directorios en diversas particiones. Pero esto puede no ser suficiente y por eso el núcleo del sistema proporciona la posibilidad de controlar el espacio de almacenamiento por grupo o usuario.

Es muy probable que muchas de las intrusiones puedan ser realizadas por los alumnos, por lo que se debe proteger al sistema de operaciones accidentales debidas a descuidos o ignorancia de parte ellos.

Una norma básica de seguridad radica en la asignación a cada usuario sólo de los permisos necesarios para poder cubrir las necesidades de su trabajo sin poner en riesgo el trabajo de los demás.

Para esto Linux dispone de todas las características de los sistemas Unix, ya que se mantiene un control de acceso a los usuarios donde se verifica login y password; por lo que se debe asegurar que los usuarios utilicen claves sólidas que cumplan con las siguientes características:

- No debe ser una palabra conocida
- Debe de contener letras, números y caracteres especiales
- Deben ser fáciles de recordar y difíciles de adivinar.

Para mantener un sistema seguro pero funcional, se tienen que otorgar los permisos adecuados para que los usuarios puedan tener acceso solo a los recursos autorizados, considerando los permisos de propietario, grupo y otros. Afortunadamente Fedora Core tiene por defecto los valores que son suficientes para mantener el sistema seguro.

Una vez que se han visto las características generales de seguridad, lo que queda es aplicar el sentido común, se tiene que ver la situación actual y responder a una serie de preguntas:

- ¿Qué se quiere proteger?
- ¿Qué valor tiene lo que se quiere proteger?
- ¿Qué costo tiene la seguridad?
- ¿De quién se quiere proteger?
- ¿Cuáles son los puntos débiles del sistema?

Después de responder estas preguntas, el administrador del sistema obtendrá un esquema elemental de la situación actual y de las medidas de seguridad básicas que debe tomar.

5.3 Trabajo futuro

Se considera que lo expuesto en esta tesis da la pauta para continuar la investigación en el área del software libre, en especial de las redes de terminales gráficas; a continuación se darán algunas ideas que pueden tomarse como base para trabajos futuros y para mejorar los resultados obtenidos:

- Se propone en primer lugar la mejora de las características del servidor para que éste pueda dar soporte a un mayor número de terminales. De acuerdo a las pruebas que se efectuaron, se pudo observar que la memoria es determinante en el rendimiento de la red, por lo que se recomienda tener 1 GB si se requiere expandir la red de 3 a 10 terminales. También es recomendable utilizar como medio de almacenamiento un disco duro SCSI en lugar de un disco IDE, ya que permite transferencias de datos con una mayor velocidad y de manera más confiable; éste aspecto es importante debido a que en un momento determinado todas las terminales pueden estar utilizando este recurso, por lo que el rendimiento de la red puede disminuir.

- Continuando con la parte de optimización de la red, se propone la implementación del proyecto PXES, que es una alternativa al LTSP, pero a diferencia de éste último permite aprovechar los recursos de las terminales con mejores capacidades, lo que

implica que se puedan correr aplicaciones localmente utilizando su memoria y disco duro, disminuyendo considerablemente la carga al servidor. Cabe mencionar que las terminales con menores capacidades como las 486, pueden seguir con el esquema del LTSP, ya que como se pudo observar en este trabajo, es el óptimo para este tipo de computadoras.

- Una posible aplicación de la red implementada en el laboratorio, es la creación de un cluster; esta arquitectura permite utilizar los recursos de las terminales para ejecutar operaciones de cálculo complejas que requieren mayor tiempo de procesamiento. El software que se propone utilizar es Open Mosix, el cual incorpora al kernel de Linux un potente algoritmo de balanceo de cargas para dividir el trabajo de procesamiento entre todos los equipos. Se debe considerar que las terminales deben poseer características de procesamiento y memoria aceptables para que el tiempo en resolver un problema específico sea mínimo.

- Otro proyecto interesante que se propone es el de la realización de una página Web que permita controlar remotamente dispositivos electrónicos conectados a las terminales, por ejemplo circuitos basados en microcontroladores. Esto permitiría entre otras cosas, monitorear el comportamiento de algunos procesos ligados a estos circuitos los cuales tienen potencialmente muchas aplicaciones. Con esto se pretende ofrecer una infraestructura que permita impulsar a los estudiantes a que desarrollen proyectos de electrónica cada vez más novedosos y funcionales.

- Finalmente, aunque es un proyecto independiente a la institución; con la experiencia adquirida en el desarrollo de esta tesis, se pretende implementar el LTSP en un cybercafe, se propone invertir en un servidor que pueda dar soporte a más de 15 computadoras 486 equipándolas con software de todo tipo, así como de Internet. La renta de los equipos de computo sería realmente mínima tomando en cuenta que pueden proporcionar los mismos resultados que si se utilizaran computadoras Pentium III, con la diferencia de brindar el mismo servicio pero a un menor precio, esto debido al ahorro de licencias, mantenimiento de los equipos, etc., y a su vez las personas se familiarizan con el uso de Linux. Posteriormente se pretende crear un servidor de música para dichas terminales, así como un servidor Samba que permita compartir recursos y archivos con diferentes plataformas de Windows (95, 98, NT, 2000, Me) y de esta manera dichas plataformas puedan acceder a los archivos Linux. El reactivar estas computadoras 486 para darle nuevas aplicaciones, y alargar su ciclo de vida; permitirá tener un negocio rentable con inversiones mínimas.

CONCLUSIONES

La idea básica del proyecto fue rescatar computadoras relativamente obsoletas como las 486 y pentium brindándoles una nueva apariencia de procesamiento gracias a la dependencia hacia un servidor, el cual les brinda sus recursos para lograr obtener unas terminales totalmente funcionales que cuentan con una buena velocidad de procesamiento, acceso a internet, procesadores de texto, software de electrónica, hasta diseños en 3D, etc., y todo bajo un ambiente gráfico totalmente amigable.

Se pudo corroborar todo esto ya que las terminales fueron probadas con éxito en el laboratorio, al principio se experimentó la facilidad de instalación del sistema operativo y una vez que se implementó el LTSP, el mantenimiento así como la actualización del software se efectuaron de una forma sencilla. El sistema realmente demostró su estabilidad y confiabilidad, nos ofrece la libertad de poder ajustarlo a nuestras necesidades, aunado a muchas otras ventajas y beneficios que se encontraron al implementar este proyecto.

También se logró recopilar una gran variedad de software de electrónica totalmente gratuito, el cual ofrece nuevas posibilidades de desarrollo para los alumnos. Además éste tipo de software permite un gran ahorro en licencias, así como de hardware adicional debido a que estas aplicaciones no necesitan de mayores recursos para funcionar adecuadamente.

Además este tipo de terminales bajo Linux no recibe tantos ataques de virus como en Windows, podemos decir que este tipo de sistema es realmente recomendable no sólo para instituciones públicas o pequeñas empresas, sino también para grandes empresas que deseen adquirir un sistema potente, seguro, y muy económico. No requiere de un gasto adicional para capacitación ya que Linux es muy intuitivo por su ambiente gráfico.

Es verdad que en Linux existen ciertos problemas con algunos modems, tarjetas de video e impresoras; donde para obtener una buena configuración se requerirá de varias horas, pero también tenemos la opción de que Linux permite ejecutar muchos programas de Windows mediante emuladores.

Con respecto al tiempo de implementación no hay mayor problema, ya que al hallar la configuración adecuada para una terminal las otras configuraciones son muy similares, esto reduce en gran medida el tiempo de implementación ya que solo es cuestión de configurar ciertas partes.

Al hacer uso de la protección de la memoria entre procesos, de manera que ninguno de ellos detenga el sistema como suele suceder usualmente en Windows, se obtiene una estabilidad determinante para el sistema así mismo cuenta con una mejor

optimización de los recursos de la computadora brindándonos la confiabilidad de realizar varios procesos sin problemas; es un sistema mucho más robusto sobre todo para aplicaciones de alta demanda.

Se optó por utilizar el proyecto LTSP debido a que nos permite aprovechar los recursos de computadoras de baja capacidad, si se contará con computadoras medianamente más potentes se pudó haber utilizado por ejemplo el proyecto de PXES ya que permite aprovechar los recursos de las terminales con mejores capacidades, lo que implica que se puedan correr aplicaciones localmente utilizando su memoria y disco duro, disminuyendo considerablemente la carga al servidor. Es por eso que promovemos el uso de un servidor más potente para lograr atender las peticiones de las terminales ya que éstas cuentan con pocos recursos, y dependen de las características del servidor.

Comprobamos que se eligió la mejor opción para reutilizar a las 486 disponibles, logrando darles nuevamente la funcionalidad que se buscaba así como nuevos campos en los que pueden ser utilizadas estas terminales bajo el concepto del LTSP.

REFERENCIAS

RAYA José Luis, RAYA Cristina, Redes Locales, Ed. Alfaomega Ra-Ma, 2002

FLORIANO Blanco Lino, GALAN Galán Susana, Al día en una hora Linux, Ed. Anaya Multimedia, 1997

CARLING M. DEGLER Stephen DENNIS James, Administración de sistemas Linux, Ed. Prentice Hall

PERRY Hekman Jessica, Linux in a nutshell. A desktop quick reference, Ed. Perry O'reilly, 1997

HUSAIN Kamran, PARKER Tim, Linux unleashed, Ed. Sams published, 1995

CARD Remy, DUMAS Erick, MEVEL Frank, The Linux kernel book, Ed. John Wiler & Suns, 1998

BLANCO J. Vicente, Linux Instalación, administración y uso del sistema, Ed. Ra-ma, 1996

NEGUS Christopher, La biblia de Red Hat Linux 7, Ed. Anaya multimedia, 2001

ESPINOSA H. Juan Carlos, Red Hat Linux 7.0 Instalación y configuración básica, Ed. Alfaomega, 2001

BENTSON Randolph, Inside Linux. A look at operating system development, Ed. Specialized System Consultants, Inc. 1996

LES Fred, DERFLER J. Jr. Frank, Guía de Novel Netware, Ed. Anaya multimedia, 1993

STAN Kelly – Bootle, Como usar Unix sistema V, versión 4.0, Ed. Noriega editores, 1993

MEDIAVILLA Manuel, Seguridad en Unix, Ed. Ra-ma, 1998

WELSH Matt, KAUFMAN Lar, Running Linux, Ed. O'reilly & associates, INC, 1996

STEFAN Strobel, VOLKER Elling, Linux unleashing the workstation in your PC, Ed. Springer, 1997

Página web de la FSF, Sección de Filosofía http://www.fsf.org/philosophy/free-sw.es.html

Página web del Proyecto GNU, Sección de Historia
http://www.gnu.org/gnu/thegnuproject.es.html

Página web del Proyecto GNU, Sección de Licencias
http://www.gnu.org/copyleft/gpl.html

Página web del Proyecto LTSP, Sección de Documentación
http://www.ltsp.org/documentation/ltsp-3.0-4-es.html

Página web de creación de imágenes de arranque http://www.rom-o-matic.net

Página web de Fedora Core http://fedora.redhat.com/download/

Página web del proyecto K12LTSP, Sección de Instalación
http://k12ltsp.org/install.html

Página web acerca de estaciones gráficas diskless
http://www.logiclinux.com/soluciones/diskless/

Página web acerca de experiencias con el LTSP http://www.cignux.org.ar/otkraus.htm

Página web de referencia para implementar un cybercafe
http://webmaster.bankhacker.com/ayuda-linux/consulta/Instalar+Cybercafe/

GLOSARIO

ASCII: (American Standard Code for Information Interchange). Es el código estándar de conjunto de caracteres que cualquier computadora puede entender, usado para representar las letras latinas, en mayúsculas, minúsculas, números, puntuación, etc. Hay 128 códigos estándar ASCII, cada uno de los cuales puede representarse por un número binario de 7 dígitos. Sin embargo, otros conjuntos de caracteres como Latin-1 están comenzando a usarse.

Broadcast: Es un tipo especial de dirección IP usada para enviar un mensaje a todos los nodos de una red.

Cliente: Programa que se usa para contactar y obtener datos de un programa de servidor localizado en otra computadora, generalmente a gran distancia. Cada programa cliente está diseñado para trabajar con uno o más tipos de programas servidores específicos, y cada servidor requiere un tipo especial de cliente.

Demonio: Es un proceso independiente encargado de llevar a cabo una tarea sin la intervención del usuario.

Dirección IP: Dirección que identifica a cada computadora en Internet utilizando una fila de cuatro conjuntos de números separados por puntos.

Diskless: PC o estación de trabajo conectado a una LAN sin disco duro ni disquetes y que depende del almacenamiento en disco de un servidor de archivos conectado a la misma LAN.

DNS: (Domain Name System). Es un sistema que traduce un nombre de dominio a un número IP, y se emplea para facilitar el manejo de direcciones ya que es más fácil recordar un nombre que un grupo de números.

EMACS: (Editing MACroS o Extensible MACro System). Es un editor muy popular para Unix y otros sistemas operativos.

Ethernet: Es la tecnología de red de área local más ampliamente utilizada. Es un estándar de la industria muy extendido, desarrollado originalmente por Xerox y formalizado en 1980 por DEC, Intel y Xerox. Las redes Ethernet transmiten datos a 10/100/1000/ Mbps utilizando un protocolo especificado.

FTP: (File Transfer Protocol). Permite transmitir archivos sobre Internet entre una máquina local y otra remota.

Host: Es una computadora anfitriona que permite a varios usuarios correr aplicaciones o programas diferentes.

Runlevel: Es el estado general de un sistema operativo tipo Unix, el cual es definido por el proceso init.

Script: Es un programa o secuencia de instrucciones que son interpretadas o ejecutadas por otro programa.

Sistema de archivos: Un sistema de archivos es el método con el que la información es introducida en un disco duro. Sistemas operativos distintos generalmente usan sistemas de archivos diferentes.

TCP/IP: (Transmisión Control Protocol/Internet Protocol) familia de protocolos que hacen posible la interconexión y trafico de red en internet.

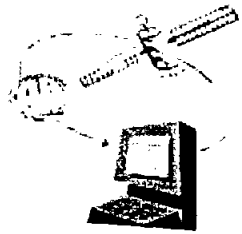
Telnet: Remote login es un protocolo que forma parte de la suit de protocolos TCP/IP y que permite conectar una terminal remota a su computadora host.

VESA: Es un estándar de modos de video para tarjetas VGA y superiores, que permite programar drivers compatibles con todas las tarjetas graficas que cumplan estas normas, independientemente del chip que incorporen.

PUESTA EN MARCHA

OBJETIVO ESPECÍFICO

Aplicar los conocimientos teóricos y prácticos para garantizar el buen funcionamiento del servidor de terminales gráficas en el momento de iniciar todo el sistema. Entender el proceso de cómo inicia una terminal para poder detectar y darle la solución más óptima a los posibles problemas.



4

INTRODUCCIÓN

Hasta este punto se han instalado los paquetes del LTSP y del software necesario, también se han realizado las configuraciones correspondientes a los archivos de sistema; sin embargo falta comprobar que efectivamente cada una de las terminales inicien una sesión gráfica debido a que estas configuraciones no garantizan que dichas terminales funcionen correctamente, para ello se cuenta con un apartado de detección y solución de problemas más comunes.

Una vez que han sido solucionados los problemas y se han iniciado las terminales gráficamente estarán listas para ser conectadas en red de forma local y a su vez tendrán acceso a la red global.

Tomando en cuenta que en el laboratorio se utiliza software comercial de electrónica, se hizo una recopilación de software de este tipo, pero que puede correr bajo Linux y con la ventaja de que es libre.

4.1 Teoría de operación

El arranque de una terminal gráfica aplicando el LTSP involucra muchos pasos, es importante conocer y comprender de forma general cada uno de ellos; ya que si en algún momento surgen problemas, estos podrían solucionarse más rápidamente, debido a que se puede deducir el origen de ellos.

El proceso de arranque de una terminal aplicando el LTSP es el siguiente:

- Cuando se enciende la terminal, ésta busca entre los dispositivos de almacenamiento disponibles un sistema operativo que iniciar; cuando detecta el código de arranque, éste se empezará a ejecutar.

- El código de inicio tratará de detectar una tarjeta de red y una vez que lo hace, la inicializa.

- Se hará una petición de broadcast buscando un servidor dhcp en la red local, esta petición incluirá la dirección MAC de la tarjeta de red.

- El proceso inetd del servidor verá la petición de broadcast e invocará al demonio dhcpd para responderla.

- El demonio dhcpd leerá el archivo de configuración, dhcpd.conf, y tratará de localizar la entrada que corresponda con la dirección MAC de la terminal. Al ser encontrada, la pondrá en un paquete de respuesta y se lo enviará a la terminal que pidió la información. Ciertos parámetros serán incluidos en este paquete, siendo los más importantes:

    - Dirección IP asignada a la terminal.
    - Máscara de subred de la red de área local.
    - Directorio home del archivo de inicio.
    - Nombre del kernel a bajar.

- El código de inicio recibirá el paquete de respuesta dhcp y configurará la interfaz TCP/IP en la tarjeta de red con los parámetros proporcionados.

- El código de inicio enviará una petición TFTP al servidor para empezar a bajar el kernel.

- Una vez que el kernel fue completamente bajado por la terminal, éste tomará el control.

- El kernel empezará a ejecutarse, inicializando todo el sistema y sus periféricos.

- El código de inicio le puede pasar información al kernel y lo hace construyendo una línea de comandos, similar a la manera en que LILO o GRUB le pasa parámetros.

- Si el kernel necesita hacer una petición dhcp, el servidor responderá con otro paquete, conteniendo la información que el kernel necesita para continuar. Los parámetros importantes en esta respuesta son:

    - Dirección IP asignada a la terminal.
    - Máscara de subred de la red de área local.
    - El directorio raíz a ser montado vía NFS.
    - La puerta de enlace (gateway).
    - El servidor de nombres (DNS).
    - El nombre de host de la terminal.

- El directorio raíz de la terminal será montado vía NFS. Este sistema de archivos será de sólo lectura; porque se pueden tener varias terminales montando el mismo sistema de archivos, y con esto se evita que en alguna de ellas se modifiquen los contenidos del mismo.

- En este punto, el control será pasado del kernel al proceso init.

- Init leerá el archivo inittab y empezará a configurar el ambiente.

- Uno de los primeros elementos de inittab es el archivo rc.local, que correrá mientras la terminal está en el estado 'sysinit'.

- El script en rc.local creará un disco RAM de 1 MB que contendrá todas las cosas que necesitan ser escritas o modificadas de alguna manera.

- Este disco RAM será montado en el directorio /tmp. Cualquier archivo que necesite ser escrito estará en el directorio /tmp, con un enlace simbólico apuntando hacia él.

- El directorio /proc es montado.

- La interfaz de loopback es configurada.

- Varios directorios son creados dentro de /tmp para que contengan algunos de los archivos transitorios que son necesarios mientras el sistema está corriendo. Algunos de ellos son:

    - /tmp/compiled
    - /tmp/var
    - /tmp/var/run

- /tmp/var/log
- /tmp/var/lock
- /tmp/var/lock/subsys

- El archivo XF86Config es generado y está basado en las entradas del archivo lts.conf. Aquí es donde la información acerca del tipo de mouse, y otros parámetros para el servidor X son combinados para crear el archivo de configuración.

- El script start_ws es creado, éste es responsable de iniciar el servidor X y cuando lo inicia, es dirigido para enviar peticiones XDMCP al servidor XDM. Esto funciona basado en la información encontrada en el archivo lts.conf.

- El archivo syslog.conf es creado, éste archivo contiene información que le indica al demonio syslogd a qué host de la red le deberá enviar la información de log. Este host de syslog está especificado en el archivo lts.conf. Hay un enlace simbólico llamado /etc/syslog.conf que apunta a /tmp/syslog.conf.

- El demonio syslogd se inicia, usando el archivo de configuración recientemente creado.

- El control es pasado otra vez a init y buscará en la entrada initdefault para saber a cuál runlevel entrar, por lo general el runlevel por defecto es 2.

- El runlevel 2 hará que se ejecute el script set_runlevel, que leerá el archivo lts.conf para saber en qué modo correr la terminal.

- Los niveles de ejecución estándar de LTSP son 3, 4 y 5.

  - El nivel 3 iniciará en un shell en la terminal.
  - El nivel 4 ejecutará una o más sesiones de telnet en el servidor. Esto es ideal para remplazar viejas terminales de tipo serial.
  - El nivel 5 entrará en el modo GUI, levantando el cliente X y haciendo una petición XDMCP al servidor, quien mandará una ventana de diálogo de inicio para permitir al usuario autenticarse. Hace falta un Display Manager corriendo en el servidor, como XDM, GDM o KDM.

4.2 Iniciando las terminales

La estrategia que se adoptó fue probar de manera independiente cada una de las terminales, para lo cual es necesario conectarlas directamente al servidor por medio de un cable cruzado; la ventaja de hacerlo de esta forma es que se tiene un mejor control en cuanto a la posible detección y solución de problemas.

Una vez que se haya comprobado que todas las terminales inician correctamente el sistema operativo y una respectiva sesión gráfica, se podrá implementar realmente la red, es decir; las terminales se conectarán al switch y éste a su vez al servidor, para lo cual será necesario utilizar cable directo en lugar de cable cruzado.

Asumiendo que las configuraciones del servidor fueron las adecuadas, al iniciar una sesión será desplegada la siguiente pantalla:

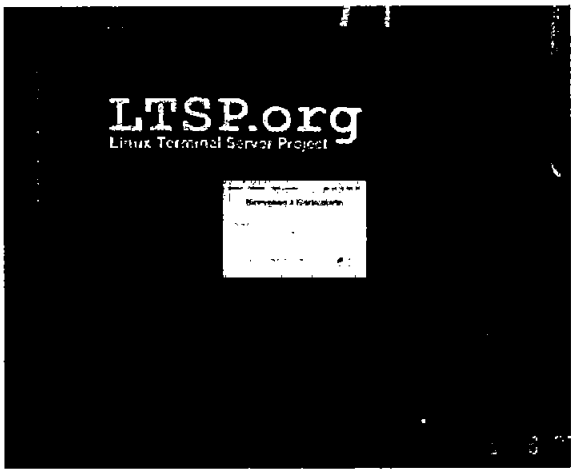


Figura 4-1 Pantalla de bienvenida del LTSP

Debido a su importancia, se describirá en la siguiente sección los problemas que se detectaron así como otros posibles errores en la implementación del LTSP, además la forma en que pueden ser solucionados.

4.3 Detección y solución de problemas

Los scripts de inicialización del LTSP hacen un gran trabajo al realizar ciertas configuraciones en los archivos necesarios, pero debido a la gran diversidad de hardware disponible para las terminales, es poco probable que éstas inicien completamente y de manera correcta la primera vez; es por eso que se debe comenzar con un proceso de detección y solución de problemas.

Conforme se va teniendo más experiencia en el proyecto LTSP, resulta más sencillo detectar los posibles errores y determinar la solución apropiada en cada caso; para ello es necesario observar hasta que punto ha llegado la terminal en su proceso de inicio.

A continuación se presentan los problemas más comunes, la manera de cómo detectarlos y sus posibles soluciones.

4.3.1 Conexiones

Es probable que la terminal no esté conectada apropiadamente a la red local, sobre todo cuando apenas se está implementando dicha red y ni siquiera se ha probado la conectividad entre las máquinas.

Para verificar esto, hay que observar que las luces de la tarjeta de red estén encendidas así como para todas las conexiones de red. Esto garantiza que al menos existe la conectividad entre la terminal y el dispositivo de la red local correspondiente.

En algunas ocasiones por cuestiones de prueba, es conveniente trabajar con la terminal conectada directamente al servidor; para ello es necesario que entre la terminal y el servidor (sin hacer uso de un hub o switch) se esté utilizando un cable cruzado.

Si se está utilizando un hub o switch, entonces hay que asegurarse que se está usando un cable directo, tanto entre las terminales y el hub, como entre el hub y el servidor.

En el capítulo 2 se muestra la configuración de los pines para construir un cable directo y un cable cruzado.

4.3.2 Imagen de arranque

La imagen de arranque debe reconocer la tarjeta de red de la terminal para poderla inicializar y configurar correctamente; si esta imagen no corresponde a la tarjeta de red, entonces no servirá para inicializar dicha terminal.

Cuando se inicia la terminal por medio de la imagen de arranque, se debe observar una salida como la siguiente:

```
loaded ROM segment 0x0800 length 0x4000 reloc 0x9400
Etherboot 5.0.1 (GPL) Tagged ELF for [LANCE/PCI]
Found AMD Lance/PCI at 0x1000, ROM address 0x0000
Probing...[LANCE/PCI] PCnet/PCI-II 79C970A base 0x1000, addr 00:50:56:81:00:01
Searching for server (DHCP)...
<sleep>
```

Tabla 4-1 Mensaje de salida cuando la imagen de arranque es apropiada

Para identificar si la imagen de arranque es la apropiada, ésta debe mostrar la dirección MAC de la tarjeta de red, sin embargo esto no garantiza que funcione completamente.

Si, por el contrario, se obtiene una salida como la siguiente, entonces es probable que la imagen de arranque generada no sea la apropiada para la tarjeta de red ya que no detecta ni muestra la dirección MAC.

```
ROM segment 0x0800 length 0x8000 reloc 0x9400
Etherboot 5.0.2 (GPL) Tagged ELF for [Tulip]
Probing...[Tulip]No adapter found
<sleep>
<abort>
```

Tabla 4-2 Mensaje de salida al iniciar la terminal con la imagen de arranque

Si no se muestran estos mensajes indicando que Etherboot a iniciado, entonces puede ser que el disquete esté dañado o que se haya grabado la imagen de arranque de manera incorrecta.

En el capítulo 3 se indica la manera de generar la imagen de arranque, así como la forma de grabarla en el disquete.

4.3.3 Problemas con el servicio DHCP

Una vez que la tarjeta de red es inicializada, el código de la imagen de arranque enviará una petición de broadcast a través de la red local, buscando algún servidor DHCP. Si la terminal obtiene una respuesta válida de parte del servidor DHCP, deberá configurar la tarjeta de red.

Si el servidor DHCP está funcionando correctamente, éste responderá a la terminal enviándole un paquete con la información que está definida en su archivo de configuración (dhcpd.conf) y que corresponde a la terminal que envió la petición.

```
ROM segment 0x0800 length 0x4000 reloc 0x9400
Etherboot 5.0.1 (GPL) Tagged ELF for [LANCE/PCI]
Found AMD Lance/PCI at 0x1000, ROM address 0x0000
Probing...[LANCE/PCI] PCnet/PCI-II 79C970A base 0x1000, addr 00:50:56:81:00:01
Searching for server (DHCP)...
<sleep>
Me: 192.168.0.1, Server: 192.168.0.254, Gateway 192.168.0.254
```

Tabla 4-3 Mensaje de que el servidor DHCP funciona correctamente

Si se muestra la línea que comienza con 'Me:', seguida por una dirección IP, entonces es un hecho que DHCP está funcionando correctamente. En ésta línea se mostrará la dirección IP asignada a la terminal, y las direcciones IP del servidor y del gateway.

Si por el contrario se muestra el siguiente mensaje en la terminal y no se obtiene respuesta por parte del servidor, es probable que haya un problema con el servicio DHCP.

```
Searching for server (DHCP)...
<sleep> <sleep>
```

Tabla 4-4 Mensaje de que quizá no funciona adecuadamente el servicio DHCP

Hay que notar que el mensaje <sleep> se puede mostrar por algunos segundos, debido al tráfico en la red, o porque por alguna razón el servidor estaba muy ocupado y no había podido atender la petición de la terminal; sin embargo no es normal que el servidor tarde mucho tiempo en dar una respuesta.

En el caso de que el servidor DHCP no responda, se tiene que averiguar las posibles causas; entre las cuales se encuentran las siguientes.

4.3.3.1 El demonio dhcpd

Para determinar si el demonio dhcpd está corriendo en el servidor se pueden ejecutar algunos comandos, e interpretar su salida; dhcpd normalmente corre en modo background escuchando en el puerto udp 67.

Se debe ejecutar el comando netstat para ver si hay algo escuchando en dicho puerto:

```
netstat -an | grep ":67 "
```

Se debe mostrar una salida similar a la siguiente:

```
udp    0      0      0.0.0.0:67      0.0.0.0:*
```

Tabla 4-5 Mensaje para saber si el demonio dhcpd esta corriendo en el servidor

La cuarta columna contiene la dirección IP y el puerto, separadas por dos puntos ":". Una dirección con solo ceros "0.0.0.0" indica que se está escuchando en todas las

interfases. Esto significa que se pueden tener varias tarjetas de red en el servidor; una interfaz eth0, otra eth1, etc., y que dhcpd esté escuchando en todas ellas.

Aunque se haya mostrado que existe un servicio escuchando en el puerto udp 67, esto no significa que efectivamente dhcpd sea el que lo esté haciendo; ya que BOOTP el cual es un servicio similar a DHCP, podría ser el que estuviera ocupando el puerto udp 67.

Actualmente en la mayoría de las distribuciones Linux solo se incluye el servicio DHCP, pero para asegurarse que el demonio esté ejecutándose realmente en el servidor, se tiene que teclear el siguiente comando:

```
ps aux | grep dhcpd
```

La salida es parecida a la siguiente:

```
root 23814 0.0 0.3 1676 820 ? S 15:13 0:00 /usr/sbin/dhcpd
root 23834 0.0 0.2 1552 600 pts/0 S 15:52 0:00 grep dhcp
```

Tabla 4-6 Mensaje para asegurarse que el demonio dhcpd esté ejecutándose

En realidad la primera línea es la que muestra que dhcpd está corriendo, la segunda es tan sólo el comando grep que se utilizó para hacer la búsqueda en la salida del comando ps.

Si no se muestra ninguna línea indicando que dhcpd está corriendo, entonces se necesita verificar que el servidor esté configurado para el runlevel 5 y que dhcpd esté listo para iniciar en dicho nivel. En sistemas basados en Fedora Core, se puede ejecutar el programa ntsysv para asegurar que el servicio DHCP esté seleccionado en el nivel de ejecución apropiado.

Existe una alternativa al comando ps, para saber si se está ejecutando dhcpd; se debe teclear lo siguiente:

```
service dhcpd status
```

Este comando mostrará el estado del servicio (si está detenido o en ejecución); si está ejecutándose indicará los ID de proceso que está utilizando.

```
Se está ejecutando dhcpd (pid 3693)...
```

Tabla 4-7 Mensaje que muestra el estado del servicio dhcpd

Si efectivamente el demonio dhcpd está detenido o con algún problema; se puede probar iniciando o reiniciando el servicio con estos comandos:

```
service dhcpd start
service dhcpd restart
```

En algunas ocasiones cuando se intenta iniciar dhcpd, aparece el siguiente mensaje:

```
st decls to leases file.
Wrote 0 new dynamic host decls to leases file.
Wrote 0 leases to leases file.
Not configured to listen on any interfaces!
```

Tabla 4-8 Mensaje de error de que no están activas las tarjetas de red

Este mensaje de error significa que no existen interfaces ethernet en el servidor o que no están activadas, esto se soluciona con el programa de configuración apropiado; en el caso de Fedora Core existe una herramienta llamada redhat-config-network, con ella se pueden configurar y activar las tarjetas de red del sistema.

Una vez que se haya(n) configurado la(s) tarjeta(s) de red, proporcionando su dirección IP, máscara de red, puertas de enlace, etc., y después de haberla(s) activado; se puede comprobar su estado con el comando ifconfig, el cual mostrará las interfaces ethernet activadas. En este momento, podrá iniciarse el demonio dhcpd sin ningún problema.

Si los comandos start o restart no arrojan errores, significa que dhcpd ahora sí, estará corriendo en el servidor.

4.3.3.2 Verificar la configuración de dhcpd.conf

Como se vio en el capítulo 3, el archivo dhcpd.conf contiene las configuraciones necesarias para que el servicio DHCP funcione correctamente; en ocasiones basta con adaptar ciertas cosas al archivo de ejemplo dhcpd.conf.example; particularmente las direcciones MAC de las terminales. Sin embargo hay situaciones en las que se debe modificar otros parámetros.

En primer lugar se debe verificar que la siguiente línea se encuentre antes del parámetro default-lease-time:

ddns-update-style none;

Esta línea es requerida a partir de la versión 3.0 de DHCP.

Hasta el momento en que la terminal necesita recibir la información para configurar la tarjeta de red y antes de comenzar a bajar el kernel desde el servidor, solo son requeridos ciertos parámetros del archivo dhcpd.conf.

Es necesario especificar las direcciones IP del servidor LTSP, del gateway y del servidor de dominio; en ambientes LTSP generalmente se utiliza un solo servidor que realice estas tres funciones, es decir una misma dirección IP será indicada en estos parámetros.

También se debe verificar que la entrada "hardware ethernet" concuerda exactamente con la dirección MAC de la terminal; ésta dirección se puede conocer cuando se inicia la terminal con la imagen de arranque apropiada.

hardware ethernet 00:E0:06:E8:00:84;

Finalmente se tienen que indicar las direcciones IP que serán asignadas a cada una de las terminales, estas se establecen en el parámetro "fixed-address".

fixed-address 192.168.0.1;

Con esta información bastará para que la terminal pueda configurar la tarjeta de red y comience a bajar la imagen del kernel desde el servidor LTSP.

4.3.3.3 Firewalls

Linux cuenta con herramientas llamadas firewalls, que disminuyen el riesgo de que algún intruso pueda comprometer la seguridad del sistema, sin embargo si estas no son configuradas adecuadamente para funcionar en conjunto con el ambiente LTSP, pueden ocasionar que el servidor no trabaje como se espera.

- Ipchains: Los kernels anteriores al 2.4 tenían la posibilidad de manipular la entrada y salida de los paquetes de datos usando ipchains, el cual utilizaba listas de reglas que se aplicaban a los paquetes en cada paso del proceso de filtrado.

Si en el servidor LTSP, se trabaja con un kernel cuya versión sea menor a la 2.4; se debe teclear el siguiente comando para determinar si ipchains está bloqueando la salida o entrada de paquetes:

```
ipchains -L –v
```

La salida debe ser similar a la siguiente:

```
Chain input (policy ACCEPT: 229714 packets, 115477216 bytes):
Chain forward (policy ACCEPT: 10 packets, 1794 bytes):
Chain output (policy ACCEPT: 188978 packets, 66087385 bytes):
```

Tabla 4-9 Mensaje que muestra si ipchains esta bloqueando los paquetes

En este caso se puede observar que ipchains está aceptando la entrada y salida de los paquetes de datos, por lo que no hay ningún conflicto entre dhcpd e ipchains.

- Iptables: Si la versión del kernel es mayor o igual a la 2.4, se tiene la posibilidad de manipular la entrada y salida de paquetes con ipchains o iptables, pero solo uno a la vez. Generalmente se utiliza iptables debido a que tiene mejoras en el funcionamiento y en el control disponible a la hora de filtrar paquetes.

Se debe ejecutar el siguiente comando para determinar el comportamiento de iptables:

```
iptables -L -v
```

La salida debe ser similar a la siguiente:

```
Chain INPUT (policy ACCEPT 18148 packets, 2623K bytes)
pkts bytes target prot opt in out source destination
Chain FORWARD (policy ACCEPT 0 packets, 0 bytes)
pkts bytes target prot opt in out source destination
Chain OUTPUT (policy ACCEPT 17721 packets, 2732K bytes)
pkts bytes target prot opt in out source destination
```

Tabla 4-10 Mensaje que muestra el comportamiento de iptables

Al igual que en ipchains, si se observa que se está aceptando la entrada y salida de paquetes, no habrá ningún problema entre iptables y dhcpd.

4.3.3.4 Problemas con la terminal

Si después de verificar las configuraciones en el servidor no se logra detectar en él algún problema, es posible que el conflicto se encuentre en la terminal. Para comprobar si la terminal está enviando la petición al servidor se puede observar el archivo /var/log/messages mientras la terminal se inicia.

Se debe teclear el siguiente comando:

```
tail -f /var/log/messages
```

Este comando mostrará las últimas líneas del archivo messages conforme se vayan agregando.

```
server dhcpd: DHCPDISCOVER from 00:50:56:81:00:01 via eth0
server dhcpd: no free leases on subnet WORKSTATIONS
server dhcpd: DHCPDISCOVER from 00:50:56:81:00:01 via eth0
server dhcpd: no free leases on subnet WORKSTATIONS
```

Tabla 4-11 Mensaje para saber si la terminal esta enviando la petición al servidor

Si se muestran algunas líneas en donde aparezca el mensaje "no free leases", entonces dhcpd está corriendo, pero en cambio hay un problema con la terminal debido a que no está haciendo la petición al servidor DHCP.

Esto puede deberse a que la imagen de arranque está defectuosa o la versión no es la correcta; aunque efectivamente dicha imagen corresponda a la tarjeta de red de la terminal, en algunas ocasiones la configuración con la que fue creada provoca que esta no funcione adecuadamente a la hora de arrancar la terminal.

Para solucionar esto, se debe probar iniciando la terminal con distintas versiones de la imagen de arranque; estas versiones se pueden conseguir también en la misma página[15].

En un caso extremo, si ninguna de las imágenes de arranque funciona, se puede probar reemplazando la tarjeta de red de la terminal; ya que puede tener algún problema físico que impida la conectividad con el servidor.

4.3.4 Problemas con el servicio TFTP

La imagen de arranque que permite iniciar las terminales contiene instrucciones o código Etherboot, el cual después de lograr comunicarse con el servidor utiliza el

[15] Página del sitio para descargar la imagen de arranque http://www.rom-o-matic.net.

protocolo TFTP para obtener un kernel Linux a través de la red. Este protocolo es muy simple, pero en ocasiones se generan ciertos problemas cuando se trata de hacerlo funcionar.

En el momento en que la terminal intenta bajar el kernel del servidor, se observa una línea similar a la siguiente:

```
Loading 192.168.0.254:/lts/vmlinuz.tulip .....
```

Tabla 4-12 Mensaje que muestra cuando la terminal intenta bajar el kernel

Posteriormente comienzan a desplegarse una serie de puntos que representan el avance de la descarga; mientras estos puntos continúen apareciendo en la pantalla significa que la imagen del kernel está siendo descargada exitosamente y por lo tanto TFTP está funcionando como se espera.

En el caso de que estos puntos no aparezcan de manera continua entonces quiere decir que hay un problema con el servicio TFTP, las posibles causas pueden ser las siguientes.

4.3.4.1 El demonio tftpd

En Fedora Core, tftp es iniciado por el demonio xinetd; existe un script de inicio llamado /etc/xinetd.d/tftp que contiene la información necesaria para que corra tftpd.

Es posible que el demonio tftpd no esté corriendo, por lo que se puede probar reiniciando el demonio xinetd, el cual reiniciará automáticamente un conjunto de servicios incluyendo tftp. Se debe teclear el siguiente comando:

```
service xinetd restart
```

4.3.4.2 La imagen del kernel no está en el directorio adecuado

La imagen del kernel que será descargado por la terminal, necesita estar en un lugar accesible para el demonio tftpd. Si la opción "-s" está especificada cuando tftpd inicia, entonces cualquier petición de la terminal debe ser relativa al directorio /tftpboot.

Por ejemplo, si el parámetro filename en el archivo de configuración dhcpd.conf es /lts/vmlinuz-2.4.19-ltsp-1, entonces la imagen del kernel deberá estar en la siguiente ruta: /tftpboot/lts/vmlinuz-2.4.19-ltsp-1.

Si no se especifica la opción "-s", entonces en el archivo dhcpd.conf tendrá que especificarse la ruta absoluta de la imagen del kernel, es decir:

```
filename        "/tftpboot/lts/vmlinuz-2.4.19-ltsp-1";
```

Tabla 4-13 Mensaje donde se especifica la ruta absoluta de la imagen del kernel

4.3.5 Problemas con el servicio NFS

NFS permite montar un sistema de archivos remoto a cada una de las terminales, éste sistema de archivos será realmente un directorio compartido que será exportado por el servidor; por lo que éste se debe configurar adecuadamente para que permita el acceso de las terminales.

Existen varios factores que pueden impedir que un sistema de archivos raíz pueda ser montado en las terminales:

4.3.5.1 El proceso Init no puede ser iniciado

Cuando la terminal intenta montar el sistema de archivos raíz, puede mostrar el siguiente error:

```
Kernel panic: No init found. Try passing init= option to kernel.
```

Tabla 4-14 Mensaje de error al intentar montar el sistema de archivos raíz

Este mensaje indica que el contenido del directorio exportado por el servidor no es el adecuado, o no contiene la información que necesita el kernel para continuar con el proceso de configuración del sistema.

Los scripts de configuración del LTSP pudieron haber tenido errores en la creación del directorio a exportar, por lo que es probable que el directorio /opt/ltsp/i386 esté vacío.

Se puede listar el contenido de este directorio para verificar que tenga la estructura correcta y la cual es la que espera encontrar el kernel de la terminal.

```
ls -l  /opt/ltsp/i386
```

Los siguientes directorios deberán estar presentes:

bin dev etc lib mnt oldroot opt proc root sbin tmp usr var

4.3.5.2 El servidor retorna el error -13

Si se muestra en la terminal el siguiente error:

```
Root-NFS: Server returned error -13 while mounting /opt/ltsp/i386
```

Tabla 4-15 Mensaje de error que muestra que el directorio i386 no esta en exports

Esto indicará que el directorio /opt/ltsp/i386 no está especificado en el archivo de configuración /etc/exports.

Es posible observar el contenido del archivo /var/log/messages para determinar alguna posible causa.

```
Jul 20 00:28:39 jamlap rpc.mountd: refused mount request from ws001
for /opt/ltsp/i386 (/): no export entry
```

Tabla 4-16 Muestra el contenido del archivo /var/log/messages

Este mensaje se puede interpretar fácilmente y confirma que efectivamente el directorio que la terminal pretende montar, no está siendo exportado por el servidor.

Para solucionar esto, basta con verificar el contenido del archivo /etc/exports y añadir si es que no existe, la siguiente línea:

```
/opt/ltsp/i386     192.168.0.0/255.255.255.0(ro,no_root_squash,sync)
```

Tabla 4-17 Muestra el contenido del archivo /

4.3.6 Problemas con el servidor X

La implementación del LTSP puede llegar a ser complicada debido a la gran diversidad de hardware, y un claro ejemplo de esto es la correcta configuración del servidor X.

Si se está utilizando una tarjeta de video y un monitor relativamente modernos, es casi seguro que estén soportados por el servidor XFree86 y por lo tanto no habrá mucho problema en que funcionen adecuadamente; sin embargo cuando se trata de equipos antiguos es necesario realizar algunos ajustes adicionales al servidor.

Los scripts de inicialización del LTSP tratarán de ejecutar el servidor X en las terminales, para que éste se comunique con el Display Manager del servidor. Si por alguna razón el servidor X no puede ser iniciado al primer intento, se probará hacerlo en 10 ocasiones más hasta que finalmente se muestre un mensaje de error si no se tuvo éxito en la conexión.

Cuando sucede esto, lo primero que se debe intentar es cambiar el servidor X especificado en el archivo lts.conf; por default se tendrá configurado el servidor XFree86 pero si la tarjeta de video es algo antigua es muy probable que se deba usar otro servidor X, por ejemplo Xvesa.

Antes de poder modificar el archivo lts.conf, será necesario instalar el paquete correspondiente al servidor Xvesa; el cual puede obtenerse en la página oficial del LTSP.

Para instalar el paquete se tiene que teclear el siguiente comando:

```
rpm –ivh  ltsp_x_vesa-3.0-1.k12ltsp.2.3.0.i386.rpm
```

Este paquete instalará los componentes necesarios para que el servidor Xvesa esté disponible para las terminales, para ello se creará el archivo Xvesa en la ruta:

```
/opt/ltsp/i386/usr/X11R6/bin/
```

Tabla 4-18 Muestra la ruta del archivo Xvesa

Una vez instalado el paquete, se requiere modificar el parámetro XSERVER en la sección de la terminal con problemas; se tendrá que cambiar el valor Xfree86 por Xvesa.

```
XSERVER = Xvesa
```

Después de efectuar los cambios correspondientes, se debe reiniciar la terminal para verificar si el servidor Xvesa soporta su tarjeta de video y monitor.

En caso de que el servidor Xvesa tampoco funcione, se tendrá que recurrir a otras alternativas, por ejemplo los servidores vga o svga; por lo cual es necesario repetir el proceso que se utilizó con Xvesa.

Hay ocasiones en que el servidor X si soporta a la tarjeta de video de la terminal, sin embargo la configuración del monitor no es la apropiada.

En modelos de monitores recientes, la configuración automática es suficiente, pero en monitores antiguos es necesario especificar algunos parámetros adicionales; los cuales definen la resolución así como la frecuencia de refresco de la pantalla.

Se tendrá que agregar los siguientes parámetros al archivo lts.conf, en la sección de la terminal:

```
X_HORZSYNC          = "30.9-49.0"
X_VERTREFRESH  = "50.0-100.0"
X_MODE_0          = 800x600
```

O bien se podrá especificar un modeline en el parámetro X_MODE_0, por ejemplo:

```
X_HORZSYNC  = "30.9-49.0"
X_VERTREFRESH = "50.0-100.0"
X_MODE_0 = 800x600  40     800  840  968 1056   600 601 605 628
    +hsync +vsync
```

Con las modificaciones antes descritas es muy probable que el servidor X inicie correctamente en cada una de las terminales, pero aun es necesario que éste logre establecer una conexión con el Display Manager que corre en el servidor.

4.3.7 Problemas con el Display Manager

El Display Manager (Administrador de Sesiones Gráficas) es un demonio que corre en el servidor, el cual espera que un servidor X se ponga en contacto con él. En el momento en que se establece la conexión se mostrará una caja de diálogo de inicio de sesión, para que el usuario pueda realmente acceder a las aplicaciones del servidor.

Los tres Display Managers más comunes en Linux son:

XDM - Forma parte del sistema estándar de X Window.
GDM - El "Gnome Display Manager" forma parte del entorno de escritorio Gnome.
KDM - El "KDE Display Manager" forma parte del entorno de escritorio KDE.

En ocasiones el servidor X logra iniciar correctamente en la terminal, pero no puede establecer una conexión con el Display Manager; cuando sucede esto, se presenta una pantalla gris con un cursor en forma de X en lugar de la caja de diálogo de inicio de sesión. Las posibles razones son las siguientes:

4.3.7.1 El Display Manager no está en ejecución

Para determinar si algún Display Manager está corriendo en el servidor se puede utilizar el siguiente comando:

```
netstat -ap | grep xdmcp
```

La salida de éste comando mostrará si hay algún proceso escuchando en el puerto xdmcp (177).

```
udp 0 0 *:xdmcp *:* 1255/gdm
```

Tabla 4-19 Muestra la salida para determinar si algún display manager esta activo

En este ejemplo se puede observar que gdm está corriendo con el PID 1255 y está escuchando en el puerto de XDMCP. En caso de que no se muestre salida alguna, significará que ningún Display Manager está corriendo; por lo cual será necesario reiniciar el sistema para reactivar el Display Manager por defecto que utiliza el servidor.

4.3.7.2 El Display Manager ignora peticiones de máquinas remotas

Si se ha comprobado que algún Display Manager está corriendo, entonces es posible que haya sido configurado para ignorar peticiones XDMCP desde máquinas remotas; para solucionar esto, es necesario verificar los archivos de configuración correspondientes al Display Manager utilizado.

4.3.7.2.1 XDM

En el archivo xdm-config la siguiente línea deberá estar comentada para que se admitan peticiones remotas:

```
#    DisplayManager.requestPort: 0
```

Y el archivo Xaccess deberá contener una línea que comience con un asterisco (*):

```
*    # Aceptar peticiones remotas
```

Ambos archivos generalmente se encuentran en la ruta /etc/X11/xdm/ pero si no es así, se pueden buscar con el comando locate.

4.3.7.2.2 KDM

El archivo kdmrc contiene una sección con la etiqueta [xdmcp], dentro de la cual hay un parámetro que deberá estar configurado de la siguiente manera:

```
Enable = true    #true habilita las peticiones remotas
```

Este archivo generalmente se encuentra en la ruta /etc/kde/kdm/ pero si no es así, se puede buscar con el comando locate.

4.3.7.2.3 GDM

El archivo gdm.conf contiene una sección con la etiqueta [xdmcp], dentro de la cual hay un parámetro que deberá estar configurado de la siguiente manera:

```
Enable = true    #true habilita las peticiones remotas
```

Este archivo generalmente se encuentra en la ruta /etc/X11/gdm/ pero si no es así, se puede buscar con el comando locate.

4.3.7.3 Conflicto del Display Manager con el DNS

Si el Display Manager definitivamente está en ejecución y está escuchando las peticiones de máquinas remotas, entonces el problema puede deberse a un simple error en el mapeo de direcciones IP a nombres, por lo que el Display Manager no sabrá con que terminal establecer la comunicación.

Para solucionar este problema es necesario configurar el archivo hosts, agregando una entrada para cada una de las terminales, por ejemplo:

```
192.168.0.1   ws001    ws001.localdomain   ws001
192.168.0.2   ws002    ws002.localdomain   ws002
```

4.3.8 Archivos de sistema

Finalmente, después de resolver los problemas detectados y realizar las modificaciones correspondientes a los archivos de configuración estos quedarán así:

```
# Archivo de configuración dhcpd.conf
ddns-update-style            none;
default-lease-time           21600;
max-lease-time               21600;
option subnet-mask           255.255.255.0;
option broadcast-address     192.168.0.255;
option routers               192.168.0.254;
option domain-name-servers   192.168.0.254;
option domain-name           "yourdomain.com";
option root-path             "192.168.0.254:/opt/ltsp/i386";
option option-128 code 128 = string;
option option-129 code 129 = text;

shared-network WORKSTATIONS {
   subnet 192.168.0.0 netmask 255.255.255.0 {
   }
}

group    {
   use-host-decl-names      on;
   option log-servers       192.168.0.254;
   host ws001 {
      hardware ethernet     00:60:8C:CA:BC:FF;
      fixed-address         192.168.0.1;
      filename              "/lts/vmlinuz-2.4.19-ltsp-1";
      option option-128     e4:45:74:68:00:00;
      option option-129     "NIC=3c509";
   }
   host ws002 {
      hardware ethernet     00:A0:24:24:F6:0F;
      fixed-address         192.168.0.2;
      filename              "/lts/vmlinuz-2.4.19-ltsp-1";
      option option-128     e4:45:74:68:00:00;
      option option-129     "NIC=3c509";
   }
   host ws003 {
      hardware ethernet     00:80:AD:3B:8F:E4;
      fixed-address         192.168.0.3;
      filename              "/lts/vmlinuz-2.4.19-ltsp-1";
   }
}
```

Tabla 4-20 Muestra le archivo completo de configuración exports

```
# Archivo de configuración exports

/opt/ltsp/i386                  192.168.0.0/255.255.255.0(ro,no_root_squash,sync)
/var/opt/ltsp/swapfiles   192.168.0.0/255.255.255.0(rw,no_root_squash,async)
```

Tabla 4-21 Muestra el archivo completo de configuración exports

```
# Archivo de configuración hosts.allow

bootpd:   0.0.0.0
in.tftpd:  192.168.0.
portmap:   192.168.0.
mountd:    192.168.0.
ALL:        127.0.0.1
ALL:      192.168.0.254
ALL:      192.168.0.1
ALL:      192.168.0.2
ALL:      192.168.0.3
```

Tabla 4-22 Muestra el archivo completo de configuración hosts.sllow

```
# Archivo de configuración lts.conf

[Default]
      SERVER          = 192.168.0.254
      XSERVER         = auto
      X_MOUSE_PROTOCOL  = "PS/2"
      X_MOUSE_DEVICE    = "/dev/psaux"
      X_MOUSE_BUTTONS   = 3
      X_USBMOUSE_PROTOCOL= "IMPS/2"
      X_USBMOUSE_DEVICE  = "/dev/input/mice"
      X_USBMOUSE_BUTTONS = 5

      XkbSymboles      = "us(pc101)"
      XkbModel         = "pc101"
      XkbLayout        = "us"

      USE_XFS          = N
      LOCAL_APPS       = N
      RUNLEVEL         = 2

      SOUND            = Y

      VOLUME           = 75

[ws001]
      XSERVER          = Xvesa
      LOCAL_APPS       = N
      USE_NFS_SWAP     = N
      SWAPFILE_SIZE    = 48m
      RUNLEVEL         = 5
      X_HORZSYNC       = "30.9-49.0"
      X_VERTREFRESH    = "50.0-100.0"
      X_MODE_0         = 800x600
[ws002]
      XSERVER          = Xvesa
      LOCAL_APPS       = N
```

```
    USE_NFS_SWAP      = N
    SWAPFILE_SIZE     = 48m
    RUNLEVEL          = 5
    X_HORZSYNC        = "30.9-49.0"
    X_VERTREFRESH     = "50.0-100.0"
    X_MODE_0          = 800x600
[ws003]
    XSERVER           = Xvesa
    LOCAL_APPS        = N
    USE_NFS_SWAP      = N
    SWAPFILE_SIZE     = 48m
    RUNLEVEL          = 5
    X_MODE_0          = 800x600
```

Tabla 4-23 Muestra el archivo completo de configuración lts.conf

4.4 Conexión en red

De acuerdo con la metodología propuesta, cuando se logra dar solución a los problemas detectados y todas las terminales inician una sesión gráfica individualmente; el siguiente paso es probar su funcionamiento conectándolas en red.

Este proceso es relativamente sencillo, pero hay que considerar algunos aspectos importantes en su implementación, los cuales básicamente son:

- Uso de cable UTP directo en lugar de cable cruzado
- Uso de un switch en lugar de un hub para garantizar un mejor rendimiento en la red
- Asignación correcta de direcciones IP para evitar conflictos entre las terminales

Tomando en cuenta los puntos anteriores, los pasos para implementar la red son los siguientes:

1. Conectar el servidor al primer puerto del switch mediante un cable directo
2. Conectar la terminal 1 al segundo puerto del switch, la segunda terminal al tercer puerto y así sucesivamente, todas estas conexiones mediante cable directo
3. Encender el servidor e iniciar los servicios necesarios del LTSP
4. Encender el switch
5. Encender cada una de las terminales y autenticarse para iniciar una sesión remota en el servidor

El siguiente diagrama muestra la configuración de los equipos al estar conectados en red:

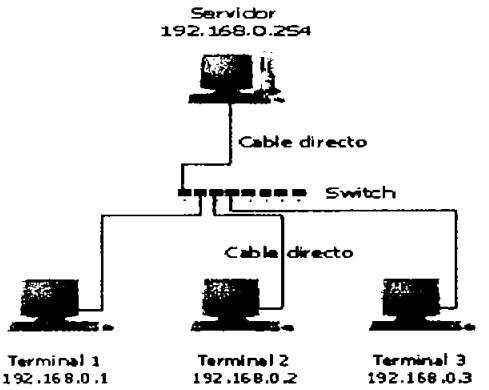


Figura 4-2 Diagrama que muestra la configuración de los equipos en red

La siguiente figura muestra a las terminales con la pantalla de bienvenida del LTSP al estar en red.

Figura 4-3 Imagen de las terminales en red con la pantalla de bienvenida del LTSP

4.5 Conexión a internet

Inicialmente, la red de terminales gráficas ha sido configurada para funcionar de manera local, esto de por si ya ofrece muchas ventajas, las cuales han sido mencionadas a lo largo de este trabajo, pero debido a que actualmente el acceso a Internet es indispensable, se optó por configurar la red local para que tenga acceso a éste.

Para establecer una conexión a Internet, debe existir un medio por el cual la red local se pueda comunicar con las redes externas; esto básicamente se logra con una tarjeta de red adicional en el servidor, lo que permite que éste también tenga la función de un gateway. Aunque es posible utilizar una sola tarjeta de red creando un "alias", esto no es recomendable debido al tráfico y a la carga que se generaría en dicho dispositivo.

Al configurar el servidor para que tenga salida a Internet por medio de la segunda tarjeta de red (eth1), automáticamente todas las terminales también tendrán acceso. Es necesario entender que en una red implementada con el LTSP, cualquier usuario conectado en un ambiente grafico, es en realidad un usuario local del servidor, por lo tanto, cualquier aplicación que lo requiera puede usar la misma conexión a Internet del servidor sin ningún inconveniente.

El programa redhat-config-network permite configurar de forma sencilla los dispositivos de red instalados en el sistema, éste puede ser ejecutado desde una terminal con el siguiente comando:

```
redhat-config-network &
```

Este programa detecta de forma automática los dispositivos de red instalados y los muestra en la sección "Dispositivos", en este caso deben aparecer dos dispositivos: eth0 y eth1.

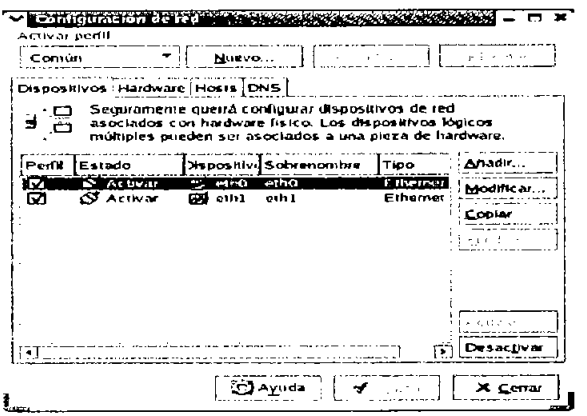


Figura 4-4Pantalla donde se muestran los dispositivos lógicos de red

En la pestaña Hardware, se debe verificar que se haya detectado correctamente la marca y modelo de las tarjetas de red; esto es necesario debido a que pueden ocurrir errores en los controladores, si estos no son los apropiados.

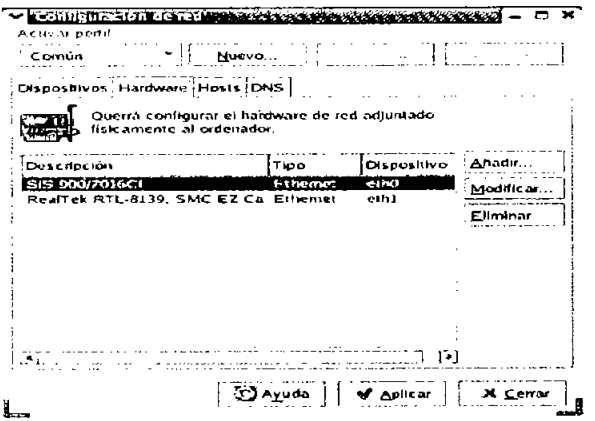


Figura 4-5 Pantalla donde se muestran los dispositivos físicos de red

La tarjeta eth0 permitirá comunicar al servidor con la red de terminales gráficas, para configurarla se debe regresar a la pestaña "Dispositivos" y seleccionar el dispositivo eth0; hacer click en "Modificar".

En la siguiente pantalla, se tiene que desactivar la opción "Obtener las configuraciones de direcciones IP automáticamente con dhcp", se activará la opción "Configurar las direcciones IP de manera estática". Se debe ingresar la dirección IP 192.168.0.254, la máscara de subred 255.255.255.0 y la puerta de enlace 192.168.0.254. Hacer click en "OK" para continuar.

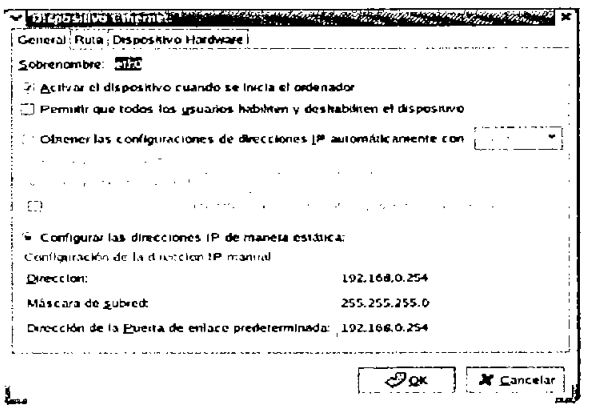


Figura 4-6 Configuración del dispositivo eth0

La tarjeta eth1 permite comunicar al servidor con las redes externas, en realidad es en este dispositivo donde se configura el acceso a Internet, por lo que se debe seleccionar y hacer click en "Modificar". En la siguiente pantalla, se tiene que desactivar la opción "Obtener las configuraciones de direcciones IP automáticamente con dhcp", se activará la opción "Configurar las direcciones IP de manera estática". Se debe ingresar la dirección IP 132.248.173.84, la máscara de subred 255.255.255.0 y la puerta de enlace 132.248.173.254. Hacer click en "OK" para continuar.

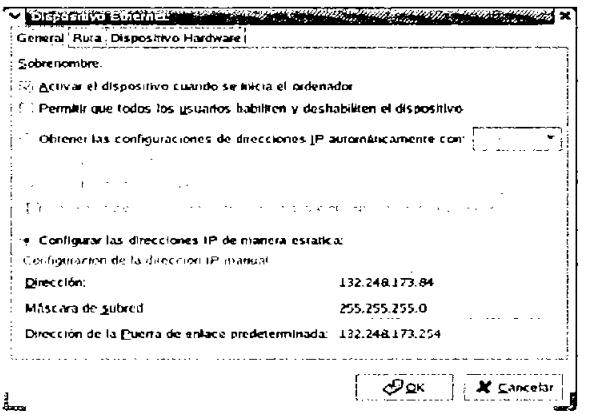


Figura 4-7 Configuración del dispositivo eth1

En la pestaña "DNS", se debe ingresar el nombre de host "Electromartin", el DNS primario 132.248.204.1 y el DNS secundario 132.248.10.2. En la sección "Ruta de búsqueda DNS", se debe añadir el dominio "aragon.unam.mx". Una vez ingresados los datos, hacer click en "Aplicar" para confirmar los cambios.

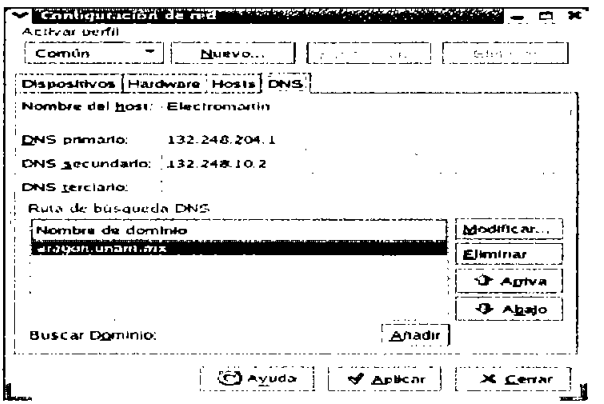


Figura 4-8 Configuración del DNS

4.6 Software específico

El LTSP es un claro ejemplo de las ventajas y posibilidades que el software libre puede ofrecer, ya que mediante la adecuada combinación de diferentes servicios y programas se logró implementar la red de terminales gráficas en el laboratorio de electrónica.

Como ya se había mencionado, dicha red proporciona entre otras cosas, un sistema centralizado de almacenamiento de información y la posibilidad de utilizar una gran variedad de aplicaciones, la mayoría de ellas distribuidas bajo la licencia GNU/GPL.

Sin embargo, no es conveniente instalar de una forma indiscriminada programas que tal vez nunca serán utilizados; para evitar esto se pensó en la necesidad de los alumnos de contar con software relacionado con la electrónica, el cual puede clasificarse en tres principales grupos: diseño, simulación y programación de circuitos.

Actualmente ya se cuenta con software de este tipo, por ejemplo: el Electronics Workbench, Pspice, Warp, Mplab, entre otros; pero debido a que estos programas corren en Windows se optó por buscar software propio de Linux que pudiera ser útil en el laboratorio y que tuviera características semejantes a los anteriores.

Cabe mencionar que el software para Windows podrá seguir utilizándose debido a que las terminales han sido configuradas para que puedan seguir arrancando este sistema operativo si así se requiere. En Internet hay muchos programas de electrónica disponibles, pero solo se describirán los que se consideraron más funcionales y amigables para el usuario.

4.6.1 Diseño

Este tipo de software es utilizado para crear diagramas o esquemas que representan gráficamente un circuito electrónico, éstos pueden imprimirse o exportarse a otros programas mediante la generación de un archivo compatible. Los programas de diseño no tienen la capacidad de simular o interpretar el funcionamiento del diagrama, sin embargo son muy útiles debido a que permiten generar fácilmente la estructura de casi cualquier circuito eléctrico - electrónico.

4.6.1.1 Eagle

Es un software de diseño de esquemas PCB (Printed Circuit Boards) con motor de autotrazado y una gran variedad de componentes. Es comercial, pero existe una licencia freeware (educacional) que limita el tamaño máximo de la placa de circuito impreso.

El programa consta de tres módulos principales:

- Editor de circuito impreso (Layout Editor)
- Editor de diagramas esquemáticos (Schematic Editor)
- Organizador de pistas (Autorouter)

Estos tres módulos están integrados en una sola interfaz de usuario, lo que permite crear fácilmente el diseño para la placa, a partir del circuito implementado en el editor esquemático.

La versión de evaluación del software permite crear circuitos impresos de hasta dos caras y con área máxima de 100 x 80 mm, suficiente para una gran variedad de circuitos sencillos.

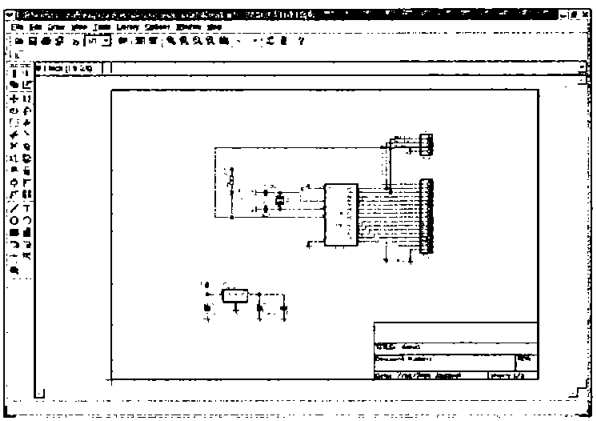


Figura 4-11 Imagen del programa Eagle

Para la instalación de Eagle, se debe descargar la última versión del código fuente[16] y seguir los pasos descritos en el capítulo 2 (métodos de instalación).

4.6.1.2 Xcircuit

Xcircuit es un programa que permite dibujar de una manera sencilla, diagramas de circuitos eléctricos de gran calidad, los cuales pueden ser exportados a otros programas como el Spice. También es posible realizar esquemas de diversas temáticas, debido a que pueden utilizarse bibliotecas de distintos objetos.

[16] Página para descargar el código http://www.cadsoft.de/info.htm

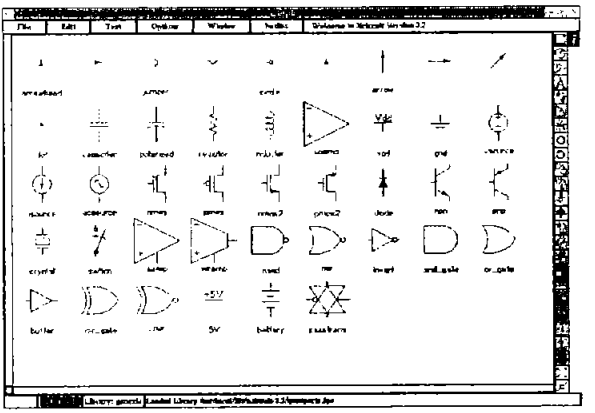


Figura 4-12 Imagen del programa Xcircuit

Para la instalación de Xcircuit, se debe descargar la última versión del código fuente[17] y seguir los pasos descritos en el capítulo 2 (métodos de instalación).

4.6.2 Simulación

Estos programas también permiten crear diagramas de circuitos electrónicos, pero tienen la gran ventaja de que se puede analizar el comportamiento físico del circuito; se puede determinar el voltaje, corriente o los estados lógicos por los que va pasando cada componente. Por lo general, los simuladores cuentan con varias herramientas entre las cuales se encuentran: los diagramas de componentes, diagramas de onda y cálculo de parámetros.

4.6.2.1 Klogic

Klogic es una herramienta para construir y simular circuitos digitales que contienen componentes estándar como AND, OR, XOR, Flip Flops RS y JK. Si se requiere construir circuitos más complejos y reutilizables se tiene la opción de diseñar subcircuitos.

Mientras se está construyendo el circuito, la simulación está ejecutándose de forma permanente, además para pruebas extendidas el programa tiene la capacidad de simular el circuito paso a paso para reconocer los estados por los que éste va pasando. También es posible desplegar la señal de flujo de los componentes de un circuito de forma gráfica e intuitiva.

[17] Página para descargar el código http://bach.ece.jhu.edu/~tim/programs/xcircuit/welcome.html

paquetes; lo que permite que su uso sea fácilmente extendido para nuevas áreas de aplicación.

Los principales componentes de Ksimus son:

- Compuertas AND, NAND, OR, NOR, XOR, XNOR
- Flip Flops RS, JK y JK-MS
- Generador de pulsos
- LED
- Latches D
- Contador
- Display de 7 segmentos
- Multiplexor y demultiplexor
- Generador de ondas
- Aritméticos (add, sub, mul, div, mod, abs)
- Trigonométricos (sin, asin, cos, acos, tan, atan, atan2)
- Hiperbólicos (sinh, asinh, cosh, acosh, tanh, atanh)
- Exponenciales (log, log10, exp, pow, square)

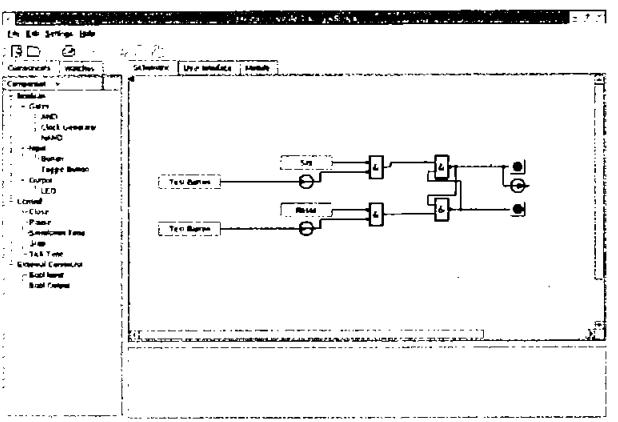


Figura 4-15 Imagen del programa Ksimus

Para la instalación de Ksimus, se debe descargar la última versión del código fuente[19] y seguir los pasos descritos en el capítulo 2 (métodos de instalación).

[19] Página para bajar el código fuente http://ksimus.berlios.de/index.html

4.6.2.3 Tkgate

Tkgate es un simulador de circuitos digitales, basado en un editor gráfico tcl/tk. Soporta una amplia gama de componentes básicos, pero además se tiene la capacidad de construir módulos mediante el diseño jerárquico de circuitos.

Los principales componentes de Tkgate son:

- Compuertas AND, NAND, OR, NOR, XOR, XNOR
- Transistores NMOS y PMOS
- Buffers triestado
- Componentes ALU (sumadores y multiplicadores)
- Elementos de memoria (registros, RAMs, ROMs)

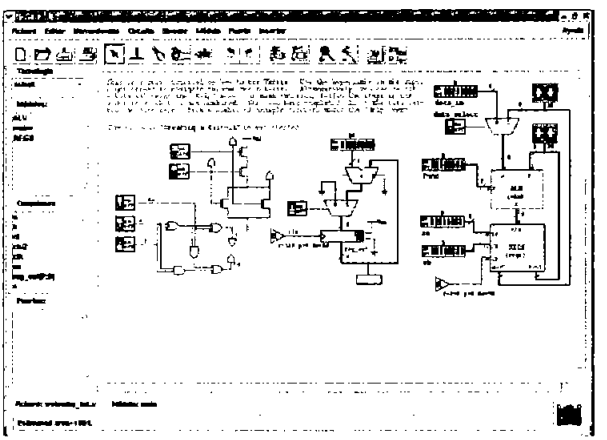


Figura 4-16 Imagen del programa Tkgate

Para la instalación de Tkgate, se debe descargar la última versión del código fuente[20] y seguir los pasos descritos en el capítulo 2 (métodos de instalación).

4.6.2.4 Vipec

Vipec es una poderosa herramienta para el análisis de circuitos de alta frecuencia y redes eléctricas lineales. Es adecuado para simular en el dominio de la frecuencia, circuitos relacionados con las comunicaciones, como: amplificadores, filtros y líneas de transmisión.

[20] Página para bajar el código fuente http://www.tkgate.org

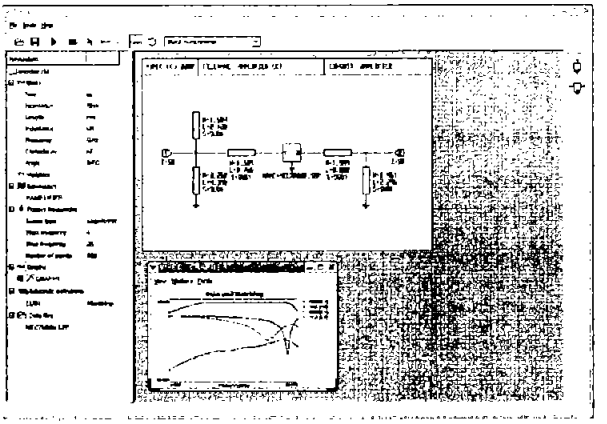


Figura 4-17 Imagen del programa Vipec

Para la instalación de Vipec, se debe descargar la última versión del código fuente[21] y seguir los pasos descritos en el capítulo 2 (métodos de instalación).

4.6.2.5 Gpsim

Este entorno de simulación para PIC's soporta microprocesadores de 12, 14 y 16 bits, además tiene la capacidad de simular en tiempo real sistemas con un ciclo de reloj de 20 MHz.

GPSIM cuenta con un depurador, un desensamblador, un manejador de estímulos, un mapa de memoria y algunos plugins que permiten simular componentes externos al PIC como una pantalla LCD.

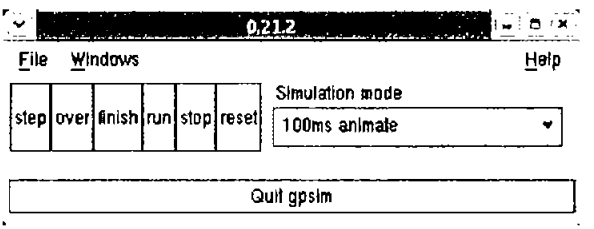


Figura 4-18 Imagen del programa Gpsim

[21] Página para bajar el código fuente http://vipec.sourceforge.net

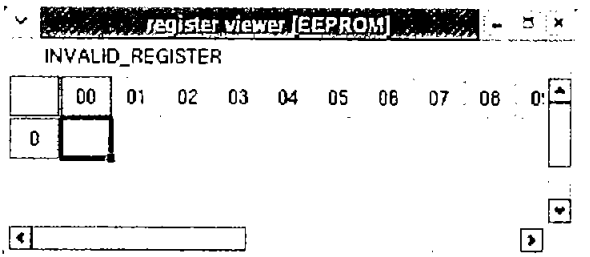


Figura 4-19 Imagen de un ejemplo de programación del programa Gpsim

Para la instalación de Gpsim, se debe descargar la última versión del código fuente[22] y seguir los pasos descritos en el capítulo 2 (métodos de instalación).

4.6.3 Programación

Estas aplicaciones permiten diseñar mediante lenguajes de alto nivel, la estructura lógica de circuitos programables, por ejemplo: memorias, microcontroladores, microprocesadores, PAL, GAL, etc.

Una vez que el diseño es terminado, éste tiene que ser copiado en el chip correspondiente, lo cual es posible utilizando un dispositivo especial denominado "grabador". En ocasiones es posible simular el comportamiento del circuito antes de ser grabado, lo que permite la optimización y depuración del diseño.

4.6.3.1 BlueHDL

BlueHDL es una suite de herramientas VHDL, que consiste en un editor y compilador VHDL, un motor de simulación, y una interfaz de visualización de ondas.

Permite diseñar con lenguajes de alto nivel logrando desarrollar circuitos programables, es una buena opción ya que para todo circuito por sencillo que sea se busca realizarlo programando un microcontrolador.

[22] Página para bajar el código fuente http://www.dattalo.com/gnupic/gpsim.html

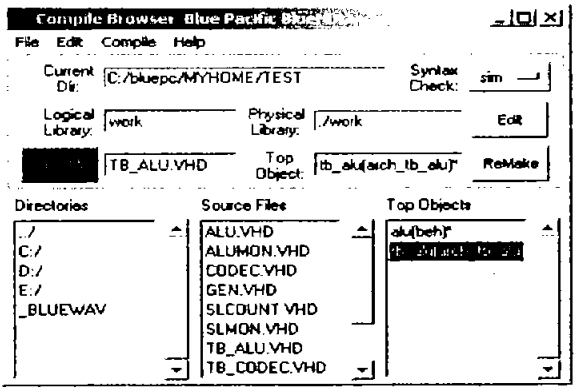


Figura 4-20 Imagen del programa BlueHDL

Para la instalación de BlueHDL, se debe descargar la última versión del código fuente[23] y seguir los pasos descritos en el capítulo 2 (métodos de instalación).

Con todas las instalaciones y configuraciones previas se comprobó que efectivamente cada una de las terminales inició una sesión gráfica adecuadamente, apoyándonos en la parte de detección y solución de problemas; además la red y la conexión a internet funcionó tal y como se esperaba. Ahora analizaremos el rendimiento del servidor para conocer cuántas terminales realmente pueden funcionar adecuadamente.

[23] Página para bajar el código fuente http://www.bluepc.com/index.html

PRUEBAS Y ANÁLISIS DE LOS RESULTADOS

OBJETIVO ESPECÍFICO

En primer lugar se comprobará el rendimiento del servidor con sus tres terminales ejecutando diversas aplicaciones, también se analizarán algunas medidas y acciones que garanticen un mínimo de seguridad y funcionalidad en la red; y por último se darán las pautas para retomar posibles proyectos relacionados o simplemente para mejorar los servicios que proporciona el servidor.

5

INTRODUCCIÓN

Para el desarrollo de este apartado se planteó la necesidad de analizar el rendimiento del servidor para tres terminales, la idea básica es dar a conocer el porcentaje de recursos que requieren las terminales para funcionar apropiadamente sin colapsarse o alentar al servidor. Los valores se obtuvieron con la ayuda del comando TOP y en el caso del procesador sus valores son dados en porcentajes para una mejor comprensión.

Por otro lado se puntualizan las ventajas de utilizar el sistema operativo Linux que proporciona algunas características con respecto a la administración y seguridad del sistema, no es necesario realizar configuraciones complejas para garantizar la seguridad que se desea, ya que el propio sistema viene con valores predefinidos que proporcionan un nivel aceptable de seguridad; pero no está de más tomar ciertas medidas y precauciones las cuales se darán en dicho apartado.

Con respecto a la administración, Linux es un sistema realmente amigable y potente, facilita el manejo de usuarios y lo más importante es que mantiene espacios restringidos para un mejor control del sistema. Es importante mencionar que la administración y la seguridad están íntimamente ligados ya que sino hubiera un control estricto del espacio de usuarios, ésto podría ser una causa para que se realizara un ataque al sistema comprometiendo la seguridad del mismo.

Una vez que se ha comprobado que el sistema es realmente estable y se tiene un control de los archivos de sistema, así como de los recursos que se le brindan a los usuarios para no permitirles un uso inapropiado de los mismos; se puede garantizar que se cuenta con una red estable, fácil de administrar y con un nivel aceptable de seguridad.

Pero eso no es todo, ya que es posible hacer mejoras o innovaciones; hay varias aplicaciones u opciones que se pueden implementar para mejorar los servicios brindados en la red de terminales gráficas.

5.1 Pruebas de rendimiento

En este apartado se probará el rendimiento del servidor y para esto se tendrá que encender terminal por terminal para tener la oportunidad de ir monitoreando las peticiones de recursos que se hacen al servidor, y a su vez tener una relación de la cantidad de memoria que se requiere para soportar n cantidad de terminales funcionando adecuadamente.

Estas pruebas ayudarán a comprobar que Linux administra eficientemente los recursos disponibles, principalmente de memoria y de procesamiento del sistema, los valores de las tablas muestran claramente cuánto se consume de los recursos del servidor pero a la vez es posible visualizar que se llega a un estado de estabilidad que impide que el sistema deje de funcionar.

El servidor cuenta con las siguientes características:

Procesador	700 Mhz
Memoria	376640 Kb
Swap	786432 Kb

Se recurrió al siguiente comando para obtener la información del uso de procesador, memoria y swap del servidor:

```
top
```

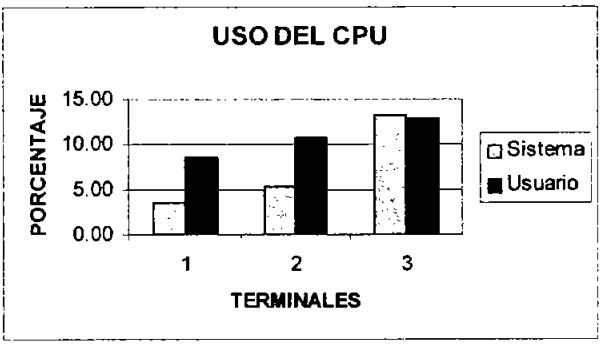
Los resultados obtenidos al momento de encender las tres terminales, se muestran en la siguiente tabla:

TERMINAL	CPU (%)		MEM (Kb)	SWAP (Kb)
	SISTEMA	USUARIO		
1	3.50	8.50	348868	0
2	5.30	10.80	364496	0
3	13.30	12.90	371024	0

Tabla 5-1 Resultados obtenidos al momento de encender las tres terminales

En la siguiente gráfica se puede notar que el procesador no tiene gran desgaste al arrancar las terminales y esto sin tomar en cuenta lo que requiere el servidor. Se puede ver que con la primer terminal se utilizó un 8.5% entonces con una segunda debería de consumir un 17% y a su vez con una tercera gastaría un 25.5% de la capacidad del procesador.

Pero esto no es así ya que realmente sólo se empleo un 13% aproximadamente y no el 25% que se tenía estimado, este comportamiento se debe principalmente a que el kernel de Linux administra eficientemente el uso del procesador; el planificador de procesos es el encargado de gestionar el uso de este recurso ya que verifica la prioridad de cada uno de los procesos y los manda a ejecución solo cuando es necesario, optimizando en gran medida los tiempos del procesador.



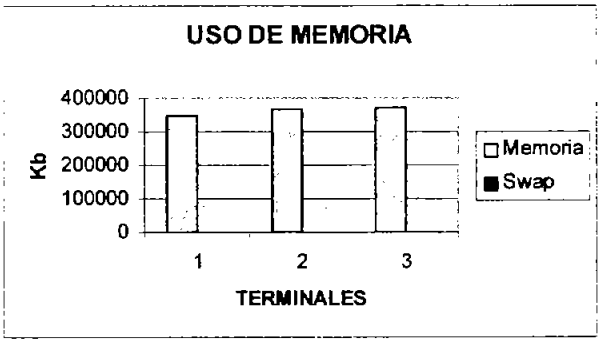
Gráfica 5-1 El uso del CPU

Se puede observar también que la memoria se está utilizando prácticamente en su totalidad, a pesar de que solo es el arranque y aún no se ha ejecutado alguna aplicación en las terminales; cabe mencionar que el sistema aun no ha requerido el uso de la memoria swap.

Bajo estas circunstancias se podría pensar que el sistema esta desperdiciando la memoria, pero no es así, ya que Linux utiliza toda la memoria libre para caché de una forma más dinámica. En un estado estable prácticamente toda la memoria principal está ocupada con fragmentos de procesos, por lo que se tendrá acceso directo a la mayor cantidad de procesos posibles disminuyendo el tiempo en la ejecución de un programa.

El kernel de Linux soporta ejecutables cargados por demanda de páginas, donde solo aquellos segmentos de un programa que se requieren son leídos del disco y puestos en la memoria.

Así, si varias instancias de un programa son ejecutadas simultáneamente, sólo una copia del código del programa estará en memoria y por esta razón el sistema se mantiene estable al ejecutar los mismos programas en las terminales.



Gráfica 5-2 El uso de la memoria

Una vez que las terminales han iniciado una sesión gráfica, es necesario verificar el rendimiento del servidor pero en este caso corriendo aplicaciones del usuario. Para esto, se eligieron 3 aplicaciones que se consideraron como las más utilizadas por los alumnos (Navegador Web, procesador de textos y hoja de cálculo), así como 3 programas específicos para el diseño y simulación de circuitos electrónicos (Xcircuit, Tkgate y Bluehdl).

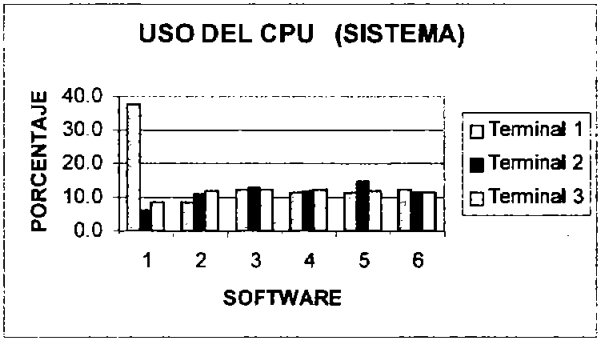
La prueba consistió en ejecutar una aplicación en la primer terminal, en este caso el navegador, luego se anotaron los valores de los recursos actuales (procesador, memoria y swap); posteriormente se hizo lo mismo con la segunda y tercer terminal. Después que se hicieron las mediciones correspondientes al primer programa, se prosiguió a ejecutar el procesador de texto, continuando con la misma mecánica se ejecutaron las aplicaciones restantes; hasta que al final se recopiló la información de cuanto fue el consumo de recursos por las tres terminales y a su vez para las 6 aplicaciones.

La siguiente tabla muestra los datos arrojados en la prueba antes descrita:

SOFTWARE	CPU (%)						MEM (Kb)			SWAP (Kb)		
	Term 1		Term 2		Term 3		Term 1	Term 2	Term 3	Term 1	Term 2	Term 3
	SIST	USR	SIST	USR	SIST	USR						
Navegador	37.6	20.5	6.0	15.9	8.7	19.7	371040	371120	365912	0	0	0
Proc. de texto	8.7	19.7	11.0	12.2	12.1	13.6	372036	371980	372216	0	0	0
Hoja de cálc.	12.2	15.2	12.9	11.6	12.5	14.7	372200	371180	371464	0	0	1132
Xcircuit	11.5	15.0	12.1	13.6	12.5	19.1	372152	372092	370964	1132	1132	1104
Tkgate	11.4	14.5	14.6	14.8	11.7	19.6	371568	371520	371508	1104	1104	1104
Bluehdl	12.4	14.8	11.4	16.8	11.2	16.1	371412	371792	371712	1104	1104	1104

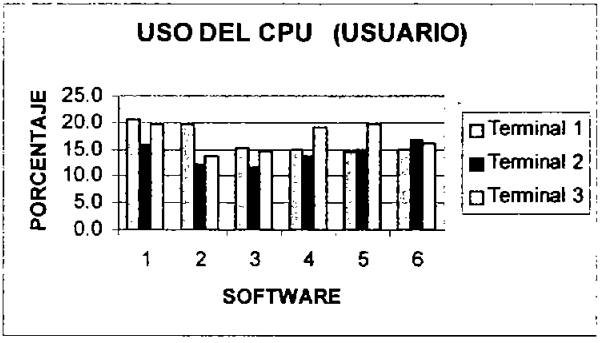
Tabla 5-2 Resultados obtenidos de las pruebas realizadas

Como puede observarse, al ejecutar el navegador en la primer terminal, se obtuvo un alto porcentaje en el uso del procesador; esto se debió a que el sistema le dio prioridad máxima a esta aplicación e incremento la actividad de procesamiento, pero después de cierto tiempo el sistema se volvió a estabilizar gracias a las características del planificador de procesos antes mencionadas.



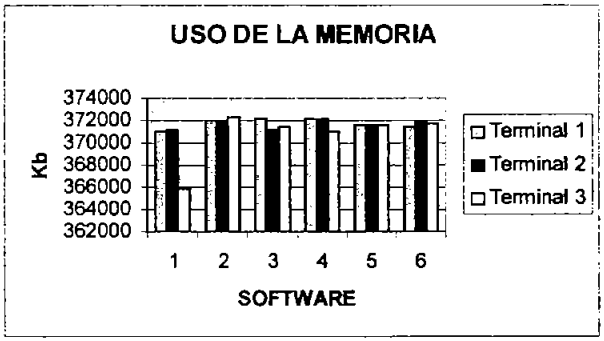
Gráfica 5-3 Uso del CPU (sistema) con las tres terminales

En el caso del uso del procesador por parte de las aplicaciones de usuario, se observa que no hubo mucha variación en el porcentaje; ya que se tiene la ventaja de que se ejecutaron las mismas aplicaciones para las tres terminales, optimizando en gran medida el uso de este dispositivo. Además con estas pruebas también fue posible comprobar la eficiencia de Linux para manejar la concurrencia en las aplicaciones.



Gráfica 5-4 Uso del CPU (usuario) con las tres terminales

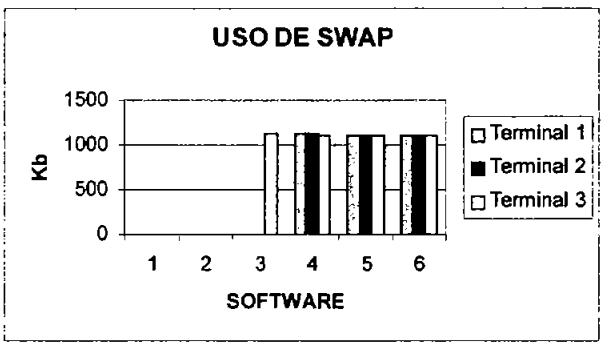
Con respecto a la memoria, se puede notar que al ejecutar cada uno de los programas ésta prácticamente se ha agotado, pero el sistema aún tiene un buen funcionamiento ya que de cierta forma llega a un punto de estabilización que impide que el sistema se colapse.



Gráfica 5-5 Uso de la memoria con las tres terminales

Sin embargo cuando la memoria física se ha terminado, Linux comienza a hacer uso de la memoria swap, sin ella las terminales se hubieran colapsado cuando la tercer terminal hiciera la petición para ejecutar el tercer programa; con esta memoria auxiliar es posible seguir trabajando y tener soporte para ejecutar tres programas más con las tres terminales funcionando adecuadamente.

La memoria swap es importante pero no puede reemplazar a una memoria física, pero puede permitir al sistema ejecutar aplicaciones más grandes guardando en disco duro aquellas partes de código que no se utilizan.



Gráfica 5-6 Uso de swap con las tres terminales

Considerando que se empezó a utilizar la memoria swap a partir del tercer programa y se cuenta con 384 MB de RAM en el servidor, se puede concluir que con un promedio de 3 aplicaciones corriendo, se necesita aproximadamente 100 MB de memoria por cada terminal; de esta manera si en un futuro se requiere expandir la red, se recomienda que por cada terminal que se incorpore, se incremente proporcionalmente el tamaño de la memoria, por ejemplo para 10 terminales se deberá tener aproximadamente 1 GB de RAM.

En cuanto al procesador, se recomienda que para más de 5 terminales, su velocidad sea mayor o igual a 1 GHz, con esto se garantiza un buen desempeño y confiabilidad de la red, solo si se desea utilizar una computadora personal como servidor.

Pero si se desea implementar este proyecto de una manera más formal, es decir, más de 15 terminales, lo más adecuado es invertir en un servidor, ya que es un equipo diseñado para estar conectado las 24 horas del día durante todo el año, el cual tiene que soportar fuertes picos de actividad, y debe tener un alto rendimiento en ejecución de aplicaciones para que los alumnos no vean afectada la velocidad y que cuente básicamente con las siguientes características, tomando como referencia las pruebas que se realizaron.

Procesador: Intel Pentium IV
Frecuencia: 3 Ghz
Memoria: 4GB
Disco Duro: 160 GB

La empresa Hewlett Packard ofrece un servidor económico de $17 000, con estas características realmente no es mucha la inversión tomando en cuenta que de esta forma se podrá tener varias 486 funcionando como estaciones de trabajo, si se invirtiera esta cantidad en computadoras medianamente potentes solo se lograría adquirir aproximadamente 4, sin tomar en cuenta la cantidad faltante por costos de licencias. Qué mejor inversión que la de adquirir este servidor y poner a funcionar esas viejas 486 brindando una potencia aceptable y sin gastos por licencias de software.

5.2 Seguridad

Con el fin de garantizar que la red de terminales gráficas funcione adecuadamente y que los recursos del servidor estén disponibles en cualquier momento, se pretende dar algunas ideas generales de los riesgos existentes y las medidas de seguridad para prevenir los posibles ataques al sistema.

La meta de la mayoría de los ataques a un sistema operativo es conseguir un acceso con privilegios de root, lo que garantiza un control total del sistema; primero se

intentará conseguir acceso como usuario normal e ir incrementando los niveles de privilegio utilizando las posibles vulnerabilidades del sistema.

Se puede decir que con el simple hecho de intentar violar la privacidad de la información, obteniendo privilegios que no le corresponden a un determinado usuario, haciendo un uso desmedido de los recursos o modificando información legítima contenida en una máquina, como puede ser el contenido de una página web o una base de datos; éstas son algunas formas de poner en riesgo el correcto funcionamiento del sistema.

En todo sistema la seguridad simplemente es fundamental, se debe tomar en cuenta que en un ambiente LTSP pueden haber muchos usuarios conectados al servidor al mismo tiempo; por lo que el sistema operativo del servidor tendrá la obligación de proteger a unos usuarios frente a otros y protegerse así mismo.

En primer lugar se debe tener en cuenta que ningún sistema es completamente seguro, solo lo será cuando no este conectado a la red o simplemente este apagado. Desde esta perspectiva se puede partir de que lo único que se puede hacer es aumentar la dificultad para que alguien pueda comprometer la seguridad del sistema.

Por tal motivo, es conveniente establecer algunas políticas que fijen el nivel de seguridad que se requiere, partiendo desde las políticas de administración del servidor, así como algunas políticas de uso para los alumnos, los cuales serán los que potencialmente podrán comprometer la seguridad de la red del laboratorio.

Lo que se desea es garantizar que el sistema permanezca funcionando de forma adecuada con comunicaciones seguras, donde nadie pueda obtener o modificar información a la que no tiene derecho. Una buena planeación ayuda bastante y antes de asegurar el sistema se debe determinar contra qué quiere protegerse y qué riesgos acepta.

Se pueden mencionar algunos requisitos de seguridad:

- Disponibilidad: Mantener la información y los recursos garantizando que no se limite el acceso autorizado a la información y el correcto funcionamiento de los recursos.
- Integridad: Requiere que la información sólo pueda ser modificada por las entidades autorizadas.
- Autenticidad: La información debe permanecer protegida ante falsificaciones, requiere de mecanismos de identificación asegurando que las comunicaciones se realizan entre entidades legítimas.
- Confidencialidad: Pretende evitar la difusión no autorizada de la información, requiere que ésta sea accesible únicamente por las entidades autorizadas.

Además en Linux se deben proteger ciertos archivos que contienen información de los usuarios (/etc/passwd, /etc/shadow), los archivos de configuración del sistema (los contenidos en etc.), el acceso al sistema y la correcta utilización de los recursos.

En muchas ocasiones los peores ataques al sistema los realiza el administrador, cuando por descuido, ignorancia o exceso de confianza altera el sistema de forma inapropiada.

Para evitar estos problemas solo hay que seguir ciertas normas de prevención como:

- No usar la cuenta de root como predeterminada, intentar primero cualquier acción como un usuario normal, si no tiene permiso puede hacer uso del comando "su" si es necesario.
- Ejecutar los comandos de forma segura verificando previamente la acción que se va a realizar.
- Activar la opción (-i) para actuar de forma interactiva con ciertos comandos que la admiten.

Al administrar un sistema Linux se deben tomar ciertas medidas que garanticen un mínimo de seguridad y funcionalidad, en primer lugar se puede limitar el espacio asignado a los usuarios ya que un posible ataque a cualquier sistema es intentar consumir todo el espacio del disco duro, una medida de seguridad es separar el árbol de directorios en diversas particiones. Pero esto puede no ser suficiente y por eso el núcleo del sistema proporciona la posibilidad de controlar el espacio de almacenamiento por grupo o usuario.

Es muy probable que muchas de las intrusiones puedan ser realizadas por los alumnos, por lo que se debe proteger al sistema de operaciones accidentales debidas a descuidos o ignorancia de parte ellos.

Una norma básica de seguridad radica en la asignación a cada usuario sólo de los permisos necesarios para poder cubrir las necesidades de su trabajo sin poner en riesgo el trabajo de los demás.

Para esto Linux dispone de todas las características de los sistemas Unix, ya que se mantiene un control de acceso a los usuarios donde se verifica login y password; por lo que se debe asegurar que los usuarios utilicen claves sólidas que cumplan con las siguientes características:

- No debe ser una palabra conocida
- Debe de contener letras, números y caracteres especiales
- Deben ser fáciles de recordar y difíciles de adivinar.

Para mantener un sistema seguro pero funcional, se tienen que otorgar los permisos adecuados para que los usuarios puedan tener acceso solo a los recursos autorizados, considerando los permisos de propietario, grupo y otros. Afortunadamente Fedora Core tiene por defecto los valores que son suficientes para mantener el sistema seguro.

Una vez que se han visto las características generales de seguridad, lo que queda es aplicar el sentido común, se tiene que ver la situación actual y responder a una serie de preguntas:

- ¿Qué se quiere proteger?
- ¿Qué valor tiene lo que se quiere proteger?
- ¿Qué costo tiene la seguridad?
- ¿De quién se quiere proteger?
- ¿Cuáles son los puntos débiles del sistema?

Después de responder estas preguntas, el administrador del sistema obtendrá un esquema elemental de la situación actual y de las medidas de seguridad básicas que debe tomar.

5.3 Trabajo futuro

Se considera que lo expuesto en esta tesis da la pauta para continuar la investigación en el área del software libre, en especial de las redes de terminales gráficas; a continuación se darán algunas ideas que pueden tomarse como base para trabajos futuros y para mejorar los resultados obtenidos:

- Se propone en primer lugar la mejora de las características del servidor para que éste pueda dar soporte a un mayor número de terminales. De acuerdo a las pruebas que se efectuaron, se pudo observar que la memoria es determinante en el rendimiento de la red, por lo que se recomienda tener 1 GB si se requiere expandir la red de 3 a 10 terminales. También es recomendable utilizar como medio de almacenamiento un disco duro SCSI en lugar de un disco IDE, ya que permite transferencias de datos con una mayor velocidad y de manera más confiable; éste aspecto es importante debido a que en un momento determinado todas las terminales pueden estar utilizando este recurso, por lo que el rendimiento de la red puede disminuir.

- Continuando con la parte de optimización de la red, se propone la implementación del proyecto PXES, que es una alternativa al LTSP, pero a diferencia de éste último permite aprovechar los recursos de las terminales con mejores capacidades, lo que

implica que se puedan correr aplicaciones localmente utilizando su memoria y disco duro, disminuyendo considerablemente la carga al servidor. Cabe mencionar que las terminales con menores capacidades como las 486, pueden seguir con el esquema del LTSP, ya que como se pudo observar en este trabajo, es el óptimo para este tipo de computadoras.

- Una posible aplicación de la red implementada en el laboratorio, es la creación de un cluster; esta arquitectura permite utilizar los recursos de las terminales para ejecutar operaciones de cálculo complejas que requieren mayor tiempo de procesamiento. El software que se propone utilizar es Open Mosix, el cual incorpora al kernel de Linux un potente algoritmo de balanceo de cargas para dividir el trabajo de procesamiento entre todos los equipos. Se debe considerar que las terminales deben poseer características de procesamiento y memoria aceptables para que el tiempo en resolver un problema específico sea mínimo.

- Otro proyecto interesante que se propone es el de la realización de una página Web que permita controlar remotamente dispositivos electrónicos conectados a las terminales, por ejemplo circuitos basados en microcontroladores. Esto permitiría entre otras cosas, monitorear el comportamiento de algunos procesos ligados a estos circuitos los cuales tienen potencialmente muchas aplicaciones. Con esto se pretende ofrecer una infraestructura que permita impulsar a los estudiantes a que desarrollen proyectos de electrónica cada vez más novedosos y funcionales.

- Finalmente, aunque es un proyecto independiente a la institución; con la experiencia adquirida en el desarrollo de esta tesis, se pretende implementar el LTSP en un cybercafe, se propone invertir en un servidor que pueda dar soporte a más de 15 computadoras 486 equipándolas con software de todo tipo, así como de Internet. La renta de los equipos de computo sería realmente mínima tomando en cuenta que pueden proporcionar los mismos resultados que si se utilizaran computadoras Pentium III, con la diferencia de brindar el mismo servicio pero a un menor precio, esto debido al ahorro de licencias, mantenimiento de los equipos, etc., y a su vez las personas se familiarizan con el uso de Linux. Posteriormente se pretende crear un servidor de música para dichas terminales, así como un servidor Samba que permita compartir recursos y archivos con diferentes plataformas de Windows (95, 98, NT, 2000, Me) y de esta manera dichas plataformas puedan acceder a los archivos Linux. El reactivar estas computadoras 486 para darle nuevas aplicaciones, y alargar su ciclo de vida; permitirá tener un negocio rentable con inversiones mínimas.

CONCLUSIONES

La idea básica del proyecto fue rescatar computadoras relativamente obsoletas como las 486 y pentium brindándoles una nueva apariencia de procesamiento gracias a la dependencia hacia un servidor, el cual les brinda sus recursos para lograr obtener unas terminales totalmente funcionales que cuentan con una buena velocidad de procesamiento, acceso a internet, procesadores de texto, software de electrónica, hasta diseños en 3D, etc., y todo bajo un ambiente gráfico totalmente amigable.

Se pudo corroborar todo esto ya que las terminales fueron probadas con éxito en el laboratorio, al principio se experimentó la facilidad de instalación del sistema operativo y una vez que se implementó el LTSP, el mantenimiento así como la actualización del software se efectuaron de una forma sencilla. El sistema realmente demostró su estabilidad y confiabilidad, nos ofrece la libertad de poder ajustarlo a nuestras necesidades, aunado a muchas otras ventajas y beneficios que se encontraron al implementar este proyecto.

También se logró recopilar una gran variedad de software de electrónica totalmente gratuito, el cual ofrece nuevas posibilidades de desarrollo para los alumnos. Además éste tipo de software permite un gran ahorro en licencias, así como de hardware adicional debido a que estas aplicaciones no necesitan de mayores recursos para funcionar adecuadamente.

Además este tipo de terminales bajo Linux no recibe tantos ataques de virus como en Windows, podemos decir que este tipo de sistema es realmente recomendable no sólo para instituciones públicas o pequeñas empresas, sino también para grandes empresas que deseen adquirir un sistema potente, seguro, y muy económico. No requiere de un gasto adicional para capacitación ya que Linux es muy intuitivo por su ambiente gráfico.

Es verdad que en Linux existen ciertos problemas con algunos modems, tarjetas de video e impresoras; donde para obtener una buena configuración se requerirá de varias horas, pero también tenemos la opción de que Linux permite ejecutar muchos programas de Windows mediante emuladores.

Con respecto al tiempo de implementación no hay mayor problema, ya que al hallar la configuración adecuada para una terminal las otras configuraciones son muy similares, esto reduce en gran medida el tiempo de implementación ya que solo es cuestión de configurar ciertas partes.

Al hacer uso de la protección de la memoria entre procesos, de manera que ninguno de ellos detenga el sistema como suele suceder usualmente en Windows, se obtiene una estabilidad determinante para el sistema así mismo cuenta con una mejor

optimización de los recursos de la computadora brindándonos la confiabilidad de realizar varios procesos sin problemas; es un sistema mucho más robusto sobre todo para aplicaciones de alta demanda.

Se optó por utilizar el proyecto LTSP debido a que nos permite aprovechar los recursos de computadoras de baja capacidad, si se contará con computadoras medianamente más potentes se pudó haber utilizado por ejemplo el proyecto de PXES ya que permite aprovechar los recursos de las terminales con mejores capacidades, lo que implica que se puedan correr aplicaciones localmente utilizando su memoria y disco duro, disminuyendo considerablemente la carga al servidor. Es por eso que promovemos el uso de un servidor más potente para lograr atender las peticiones de las terminales ya que éstas cuentan con pocos recursos, y dependen de las características del servidor.

Comprobamos que se eligió la mejor opción para reutilizar a las 486 disponibles, logrando darles nuevamente la funcionalidad que se buscaba así como nuevos campos en los que pueden ser utilizadas estas terminales bajo el concepto del LTSP.

REFERENCIAS

RAYA José Luis, RAYA Cristina, Redes Locales, Ed. Alfaomega Ra-Ma, 2002

FLORIANO Blanco Lino, GALAN Galán Susana, Al día en una hora Linux, Ed. Anaya Multimedia, 1997

CARLING M. DEGLER Stephen DENNIS James, Administración de sistemas Linux, Ed. Prentice Hall

PERRY Hekman Jessica, Linux in a nutshell. A desktop quick reference, Ed. Perry O'reilly, 1997

HUSAIN Kamran, PARKER Tim, Linux unleashed, Ed. Sams published, 1995

CARD Remy, DUMAS Erick, MEVEL Frank, The Linux kernel book, Ed. John Wiler & Suns, 1998

BLANCO J. Vicente, Linux Instalación, administración y uso del sistema, Ed. Ra-ma, 1996

NEGUS Christopher, La biblia de Red Hat Linux 7, Ed. Anaya multimedia, 2001

ESPINOSA H. Juan Carlos, Red Hat Linux 7.0 Instalación y configuración básica, Ed. Alfaomega, 2001

BENTSON Randolph, Inside Linux. A look at operating system development, Ed. Specialized System Consultants, Inc. 1996

LES Fred, DERFLER J. Jr. Frank, Guía de Novel Netware, Ed. Anaya multimedia, 1993

STAN Kelly – Bootle, Como usar Unix sistema V, versión 4.0, Ed. Noriega editores, 1993

MEDIAVILLA Manuel, Seguridad en Unix, Ed. Ra-ma, 1998

WELSH Matt, KAUFMAN Lar, Running Linux, Ed. O'reilly & associates, INC, 1996

STEFAN Strobel, VOLKER Elling, Linux unleashing the workstation in your PC, Ed. Springer, 1997

Página web de la FSF, Sección de Filosofía http://www.fsf.org/philosophy/free-sw.es.html

Página web del Proyecto GNU, Sección de Historia
http://www.gnu.org/gnu/thegnuproject.es.html

Página web del Proyecto GNU, Sección de Licencias
http://www.gnu.org/copyleft/gpl.html

Página web del Proyecto LTSP, Sección de Documentación
http://www.ltsp.org/documentation/ltsp-3.0-4-es.html

Página web de creación de imágenes de arranque http://www.rom-o-matic.net

Página web de Fedora Core http://fedora.redhat.com/download/

Página web del proyecto K12LTSP, Sección de Instalación
http://k12ltsp.org/install.html

Página web acerca de estaciones gráficas diskless
http://www.logiclinux.com/soluciones/diskless/

Página web acerca de experiencias con el LTSP http://www.cignux.org.ar/otkraus.htm

Página web de referencia para implementar un cybercafe
http://webmaster.bankhacker.com/ayuda-linux/consulta/Instalar+Cybercafe/

GLOSARIO

ASCII: (American Standard Code for Information Interchange). Es el código estándar de conjunto de caracteres que cualquier computadora puede entender, usado para representar las letras latinas, en mayúsculas, minúsculas, números, puntuación, etc. Hay 128 códigos estándar ASCII, cada uno de los cuales puede representarse por un número binario de 7 dígitos. Sin embargo, otros conjuntos de caracteres como Latin-1 están comenzando a usarse.

Broadcast: Es un tipo especial de dirección IP usada para enviar un mensaje a todos los nodos de una red.

Cliente: Programa que se usa para contactar y obtener datos de un programa de servidor localizado en otra computadora, generalmente a gran distancia. Cada programa cliente está diseñado para trabajar con uno o más tipos de programas servidores específicos, y cada servidor requiere un tipo especial de cliente.

Demonio: Es un proceso independiente encargado de llevar a cabo una tarea sin la intervención del usuario.

Dirección IP: Dirección que identifica a cada computadora en Internet utilizando una fila de cuatro conjuntos de números separados por puntos.

Diskless: PC o estación de trabajo conectado a una LAN sin disco duro ni disquetes y que depende del almacenamiento en disco de un servidor de archivos conectado a la misma LAN.

DNS: (Domain Name System). Es un sistema que traduce un nombre de dominio a un número IP, y se emplea para facilitar el manejo de direcciones ya que es más fácil recordar un nombre que un grupo de números.

EMACS: (Editing MACroS o Extensible MACro System). Es un editor muy popular para Unix y otros sistemas operativos.

Ethernet: Es la tecnología de red de área local más ampliamente utilizada. Es un estándar de la industria muy extendido, desarrollado originalmente por Xerox y formalizado en 1980 por DEC, Intel y Xerox. Las redes Ethernet transmiten datos a 10/100/1000/ Mbps utilizando un protocolo especificado.

FTP: (File Transfer Protocol). Permite transmitir archivos sobre Internet entre una máquina local y otra remota.

Host: Es una computadora anfitriona que permite a varios usuarios correr aplicaciones o programas diferentes.

Runlevel: Es el estado general de un sistema operativo tipo Unix, el cual es definido por el proceso init.

Script: Es un programa o secuencia de instrucciones que son interpretadas o ejecutadas por otro programa.

Sistema de archivos: Un sistema de archivos es el método con el que la información es introducida en un disco duro. Sistemas operativos distintos generalmente usan sistemas de archivos diferentes.

TCP/IP: (Transmisión Control Protocol/Internet Protocol) familia de protocolos que hacen posible la interconexión y trafico de red en internet.

Telnet: Remote login es un protocolo que forma parte de la suit de protocolos TCP/IP y que permite conectar una terminal remota a su computadora host.

VESA: Es un estándar de modos de video para tarjetas VGA y superiores, que permite programar drivers compatibles con todas las tarjetas graficas que cumplan estas normas, independientemente del chip que incorporen.

PRUEBAS Y ANÁLISIS DE LOS RESULTADOS

OBJETIVO ESPECÍFICO

En primer lugar se comprobará el rendimiento del servidor con sus tres terminales ejecutando diversas aplicaciones, también se analizarán algunas medidas y acciones que garanticen un mínimo de seguridad y funcionalidad en la red; y por último se darán las pautas para retomar posibles proyectos relacionados o simplemente para mejorar los servicios que proporciona el servidor.

5

INTRODUCCIÓN

Para el desarrollo de este apartado se planteó la necesidad de analizar el rendimiento del servidor para tres terminales, la idea básica es dar a conocer el porcentaje de recursos que requieren las terminales para funcionar apropiadamente sin colapsarse o alentar al servidor. Los valores se obtuvieron con la ayuda del comando TOP y en el caso del procesador sus valores son dados en porcentajes para una mejor comprensión.

Por otro lado se puntualizan las ventajas de utilizar el sistema operativo Linux que proporciona algunas características con respecto a la administración y seguridad del sistema, no es necesario realizar configuraciones complejas para garantizar la seguridad que se desea, ya que el propio sistema viene con valores predefinidos que proporcionan un nivel aceptable de seguridad; pero no está de más tomar ciertas medidas y precauciones las cuales se darán en dicho apartado.

Con respecto a la administración, Linux es un sistema realmente amigable y potente, facilita el manejo de usuarios y lo más importante es que mantiene espacios restringidos para un mejor control del sistema. Es importante mencionar que la administración y la seguridad están íntimamente ligados ya que sino hubiera un control estricto del espacio de usuarios, ésto podría ser una causa para que se realizara un ataque al sistema comprometiendo la seguridad del mismo.

Una vez que se ha comprobado que el sistema es realmente estable y se tiene un control de los archivos de sistema, así como de los recursos que se le brindan a los usuarios para no permitirles un uso inapropiado de los mismos; se puede garantizar que se cuenta con una red estable, fácil de administrar y con un nivel aceptable de seguridad.

Pero eso no es todo, ya que es posible hacer mejoras o innovaciones; hay varias aplicaciones u opciones que se pueden implementar para mejorar los servicios brindados en la red de terminales gráficas.

5.1 Pruebas de rendimiento

En este apartado se probará el rendimiento del servidor y para esto se tendrá que encender terminal por terminal para tener la oportunidad de ir monitoreando las peticiones de recursos que se hacen al servidor, y a su vez tener una relación de la cantidad de memoria que se requiere para soportar n cantidad de terminales funcionando adecuadamente.

Estas pruebas ayudarán a comprobar que Linux administra eficientemente los recursos disponibles, principalmente de memoria y de procesamiento del sistema, los valores de las tablas muestran claramente cuánto se consume de los recursos del servidor pero a la vez es posible visualizar que se llega a un estado de estabilidad que impide que el sistema deje de funcionar.

El servidor cuenta con las siguientes características:

Procesador	700 Mhz
Memoria	376640 Kb
Swap	786432 Kb

Se recurrió al siguiente comando para obtener la información del uso de procesador, memoria y swap del servidor:

```
top
```

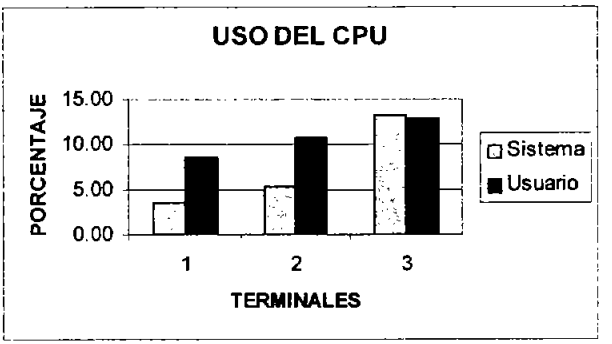
Los resultados obtenidos al momento de encender las tres terminales, se muestran en la siguiente tabla:

TERMINAL	CPU (%)		MEM (Kb)	SWAP (Kb)
	SISTEMA	USUARIO		
1	3.50	8.50	348868	0
2	5.30	10.80	364496	0
3	13.30	12.90	371024	0

Tabla 5-1 Resultados obtenidos al momento de encender las tres terminales

En la siguiente gráfica se puede notar que el procesador no tiene gran desgaste al arrancar las terminales y esto sin tomar en cuenta lo que requiere el servidor. Se puede ver que con la primer terminal se utilizó un 8.5% entonces con una segunda debería de consumir un 17% y a su vez con una tercera gastaría un 25.5% de la capacidad del procesador.

Pero esto no es así ya que realmente sólo se empleo un 13% aproximadamente y no el 25% que se tenía estimado, este comportamiento se debe principalmente a que el kernel de Linux administra eficientemente el uso del procesador; el planificador de procesos es el encargado de gestionar el uso de este recurso ya que verifica la prioridad de cada uno de los procesos y los manda a ejecución solo cuando es necesario, optimizando en gran medida los tiempos del procesador.



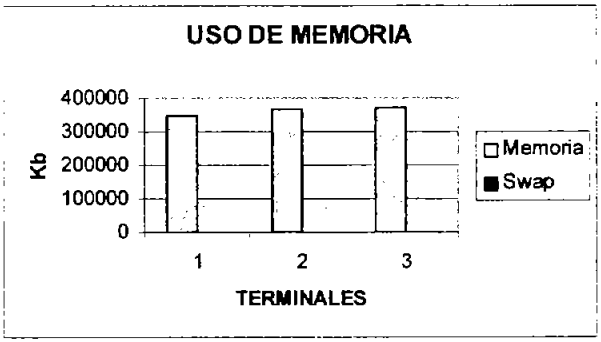
Gráfica 5-1 El uso del CPU

Se puede observar también que la memoria se está utilizando prácticamente en su totalidad, a pesar de que solo es el arranque y aún no se ha ejecutado alguna aplicación en las terminales; cabe mencionar que el sistema aun no ha requerido el uso de la memoria swap.

Bajo estas circunstancias se podría pensar que el sistema esta desperdiciando la memoria, pero no es así, ya que Linux utiliza toda la memoria libre para caché de una forma más dinámica. En un estado estable prácticamente toda la memoria principal está ocupada con fragmentos de procesos, por lo que se tendrá acceso directo a la mayor cantidad de procesos posibles disminuyendo el tiempo en la ejecución de un programa.

El kernel de Linux soporta ejecutables cargados por demanda de páginas, donde solo aquellos segmentos de un programa que se requieren son leídos del disco y puestos en la memoria.

Así, si varias instancias de un programa son ejecutadas simultáneamente, sólo una copia del código del programa estará en memoria y por esta razón el sistema se mantiene estable al ejecutar los mismos programas en las terminales.



Gráfica 5-2 El uso de la memoria

Una vez que las terminales han iniciado una sesión gráfica, es necesario verificar el rendimiento del servidor pero en este caso corriendo aplicaciones del usuario. Para esto, se eligieron 3 aplicaciones que se consideraron como las más utilizadas por los alumnos (Navegador Web, procesador de textos y hoja de cálculo), así como 3 programas específicos para el diseño y simulación de circuitos electrónicos (Xcircuit, Tkgate y Bluehdl).

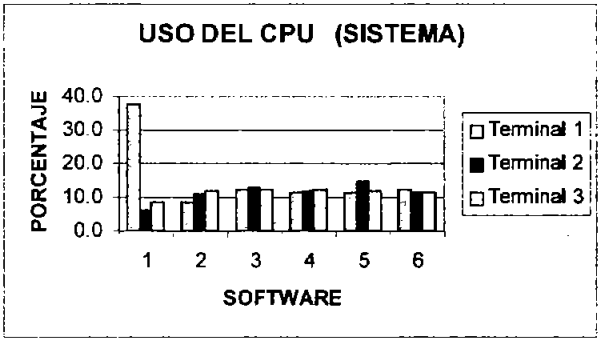
La prueba consistió en ejecutar una aplicación en la primer terminal, en este caso el navegador, luego se anotaron los valores de los recursos actuales (procesador, memoria y swap); posteriormente se hizo lo mismo con la segunda y tercer terminal. Después que se hicieron las mediciones correspondientes al primer programa, se prosiguió a ejecutar el procesador de texto, continuando con la misma mecánica se ejecutaron las aplicaciones restantes; hasta que al final se recopiló la información de cuanto fue el consumo de recursos por las tres terminales y a su vez para las 6 aplicaciones.

La siguiente tabla muestra los datos arrojados en la prueba antes descrita:

SOFTWARE	CPU (%)						MEM (Kb)			SWAP (Kb)		
	Term 1		Term 2		Term 3		Term 1	Term 2	Term 3	Term 1	Term 2	Term 3
	SIST	USR	SIST	USR	SIST	USR						
Navegador	37.6	20.5	6.0	15.9	8.7	19.7	371040	371120	365912	0	0	0
Proc. de texto	8.7	19.7	11.0	12.2	12.1	13.6	372036	371980	372216	0	0	0
Hoja de cálc.	12.2	15.2	12.9	11.6	12.5	14.7	372200	371180	371464	0	0	1132
Xcircuit	11.5	15.0	12.1	13.6	12.5	19.1	372152	372092	370964	1132	1132	1104
Tkgate	11.4	14.5	14.6	14.8	11.7	19.6	371568	371520	371508	1104	1104	1104
Bluehdl	12.4	14.8	11.4	16.8	11.2	16.1	371412	371792	371712	1104	1104	1104

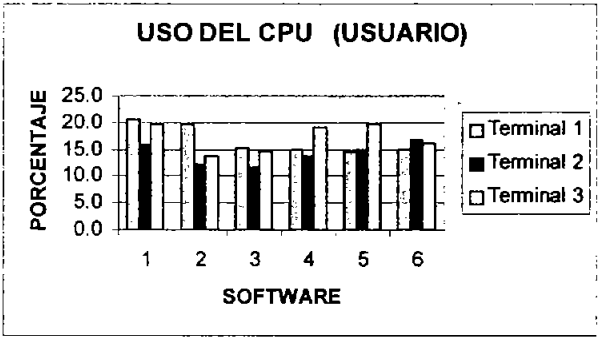
Tabla 5-2 Resultados obtenidos de las pruebas realizadas

Como puede observarse, al ejecutar el navegador en la primer terminal, se obtuvo un alto porcentaje en el uso del procesador; esto se debió a que el sistema le dio prioridad máxima a esta aplicación e incremento la actividad de procesamiento, pero después de cierto tiempo el sistema se volvió a estabilizar gracias a las características del planificador de procesos antes mencionadas.



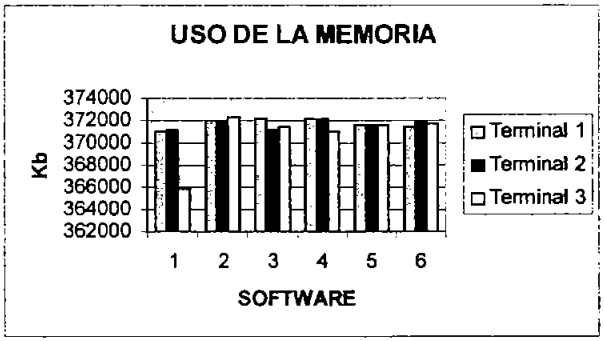
Gráfica 5-3 Uso del CPU (sistema) con las tres terminales

En el caso del uso del procesador por parte de las aplicaciones de usuario, se observa que no hubo mucha variación en el porcentaje; ya que se tiene la ventaja de que se ejecutaron las mismas aplicaciones para las tres terminales, optimizando en gran medida el uso de este dispositivo. Además con estas pruebas también fue posible comprobar la eficiencia de Linux para manejar la concurrencia en las aplicaciones.



Gráfica 5-4 Uso del CPU (usuario) con las tres terminales

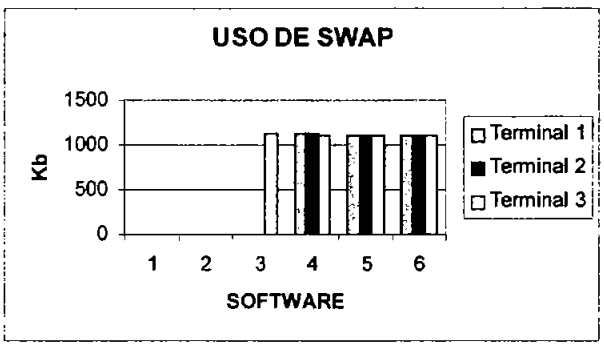
Con respecto a la memoria, se puede notar que al ejecutar cada uno de los programas ésta prácticamente se ha agotado, pero el sistema aún tiene un buen funcionamiento ya que de cierta forma llega a un punto de estabilización que impide que el sistema se colapse.



Gráfica 5-5 Uso de la memoria con las tres terminales

Sin embargo cuando la memoria física se ha terminado, Linux comienza a hacer uso de la memoria swap, sin ella las terminales se hubieran colapsado cuando la tercer terminal hiciera la petición para ejecutar el tercer programa; con esta memoria auxiliar es posible seguir trabajando y tener soporte para ejecutar tres programas más con las tres terminales funcionando adecuadamente.

La memoria swap es importante pero no puede reemplazar a una memoria física, pero puede permitir al sistema ejecutar aplicaciones más grandes guardando en disco duro aquellas partes de código que no se utilizan.



Gráfica 5-6 Uso de swap con las tres terminales

Considerando que se empezó a utilizar la memoria swap a partir del tercer programa y se cuenta con 384 MB de RAM en el servidor, se puede concluir que con un promedio de 3 aplicaciones corriendo, se necesita aproximadamente 100 MB de memoria por cada terminal; de esta manera si en un futuro se requiere expandir la red, se recomienda que por cada terminal que se incorpore, se incremente proporcionalmente el tamaño de la memoria, por ejemplo para 10 terminales se deberá tener aproximadamente 1 GB de RAM.

En cuanto al procesador, se recomienda que para más de 5 terminales, su velocidad sea mayor o igual a 1 GHz, con esto se garantiza un buen desempeño y confiabilidad de la red, solo si se desea utilizar una computadora personal como servidor.

Pero si se desea implementar este proyecto de una manera más formal, es decir, más de 15 terminales, lo más adecuado es invertir en un servidor, ya que es un equipo diseñado para estar conectado las 24 horas del día durante todo el año, el cual tiene que soportar fuertes picos de actividad, y debe tener un alto rendimiento en ejecución de aplicaciones para que los alumnos no vean afectada la velocidad y que cuente básicamente con las siguientes características, tomando como referencia las pruebas que se realizaron.

Procesador: Intel Pentium IV
Frecuencia: 3 Ghz
Memoria: 4GB
Disco Duro: 160 GB

La empresa Hewlett Packard ofrece un servidor económico de $17 000, con estas características realmente no es mucha la inversión tomando en cuenta que de esta forma se podrá tener varias 486 funcionando como estaciones de trabajo, si se invirtiera esta cantidad en computadoras medianamente potentes solo se lograría adquirir aproximadamente 4, sin tomar en cuenta la cantidad faltante por costos de licencias. Qué mejor inversión que la de adquirir este servidor y poner a funcionar esas viejas 486 brindando una potencia aceptable y sin gastos por licencias de software.

5.2 Seguridad

Con el fin de garantizar que la red de terminales gráficas funcione adecuadamente y que los recursos del servidor estén disponibles en cualquier momento, se pretende dar algunas ideas generales de los riesgos existentes y las medidas de seguridad para prevenir los posibles ataques al sistema.

La meta de la mayoría de los ataques a un sistema operativo es conseguir un acceso con privilegios de root, lo que garantiza un control total del sistema; primero se

intentará conseguir acceso como usuario normal e ir incrementando los niveles de privilegio utilizando las posibles vulnerabilidades del sistema.

Se puede decir que con el simple hecho de intentar violar la privacidad de la información, obteniendo privilegios que no le corresponden a un determinado usuario, haciendo un uso desmedido de los recursos o modificando información legítima contenida en una máquina, como puede ser el contenido de una página web o una base de datos; éstas son algunas formas de poner en riesgo el correcto funcionamiento del sistema.

En todo sistema la seguridad simplemente es fundamental, se debe tomar en cuenta que en un ambiente LTSP pueden haber muchos usuarios conectados al servidor al mismo tiempo; por lo que el sistema operativo del servidor tendrá la obligación de proteger a unos usuarios frente a otros y protegerse así mismo.

En primer lugar se debe tener en cuenta que ningún sistema es completamente seguro, solo lo será cuando no este conectado a la red o simplemente este apagado. Desde esta perspectiva se puede partir de que lo único que se puede hacer es aumentar la dificultad para que alguien pueda comprometer la seguridad del sistema.

Por tal motivo, es conveniente establecer algunas políticas que fijen el nivel de seguridad que se requiere, partiendo desde las políticas de administración del servidor, así como algunas políticas de uso para los alumnos, los cuales serán los que potencialmente podrán comprometer la seguridad de la red del laboratorio.

Lo que se desea es garantizar que el sistema permanezca funcionando de forma adecuada con comunicaciones seguras, donde nadie pueda obtener o modificar información a la que no tiene derecho. Una buena planeación ayuda bastante y antes de asegurar el sistema se debe determinar contra qué quiere protegerse y qué riesgos acepta.

Se pueden mencionar algunos requisitos de seguridad:

- Disponibilidad: Mantener la información y los recursos garantizando que no se limite el acceso autorizado a la información y el correcto funcionamiento de los recursos.
- Integridad: Requiere que la información sólo pueda ser modificada por las entidades autorizadas.
- Autenticidad: La información debe permanecer protegida ante falsificaciones, requiere de mecanismos de identificación asegurando que las comunicaciones se realizan entre entidades legítimas.
- Confidencialidad: Pretende evitar la difusión no autorizada de la información, requiere que ésta sea accesible únicamente por las entidades autorizadas.

Además en Linux se deben proteger ciertos archivos que contienen información de los usuarios (/etc/passwd, /etc/shadow), los archivos de configuración del sistema (los contenidos en etc.), el acceso al sistema y la correcta utilización de los recursos.

En muchas ocasiones los peores ataques al sistema los realiza el administrador, cuando por descuido, ignorancia o exceso de confianza altera el sistema de forma inapropiada.

Para evitar estos problemas solo hay que seguir ciertas normas de prevención como:

- No usar la cuenta de root como predeterminada, intentar primero cualquier acción como un usuario normal, si no tiene permiso puede hacer uso del comando "su" si es necesario.
- Ejecutar los comandos de forma segura verificando previamente la acción que se va a realizar.
- Activar la opción (-i) para actuar de forma interactiva con ciertos comandos que la admiten.

Al administrar un sistema Linux se deben tomar ciertas medidas que garanticen un mínimo de seguridad y funcionalidad, en primer lugar se puede limitar el espacio asignado a los usuarios ya que un posible ataque a cualquier sistema es intentar consumir todo el espacio del disco duro, una medida de seguridad es separar el árbol de directorios en diversas particiones. Pero esto puede no ser suficiente y por eso el núcleo del sistema proporciona la posibilidad de controlar el espacio de almacenamiento por grupo o usuario.

Es muy probable que muchas de las intrusiones puedan ser realizadas por los alumnos, por lo que se debe proteger al sistema de operaciones accidentales debidas a descuidos o ignorancia de parte ellos.

Una norma básica de seguridad radica en la asignación a cada usuario sólo de los permisos necesarios para poder cubrir las necesidades de su trabajo sin poner en riesgo el trabajo de los demás.

Para esto Linux dispone de todas las características de los sistemas Unix, ya que se mantiene un control de acceso a los usuarios donde se verifica login y password; por lo que se debe asegurar que los usuarios utilicen claves sólidas que cumplan con las siguientes características:

- No debe ser una palabra conocida
- Debe de contener letras, números y caracteres especiales
- Deben ser fáciles de recordar y difíciles de adivinar.

Para mantener un sistema seguro pero funcional, se tienen que otorgar los permisos adecuados para que los usuarios puedan tener acceso solo a los recursos autorizados, considerando los permisos de propietario, grupo y otros. Afortunadamente Fedora Core tiene por defecto los valores que son suficientes para mantener el sistema seguro.

Una vez que se han visto las características generales de seguridad, lo que queda es aplicar el sentido común, se tiene que ver la situación actual y responder a una serie de preguntas:

- ¿Qué se quiere proteger?
- ¿Qué valor tiene lo que se quiere proteger?
- ¿Qué costo tiene la seguridad?
- ¿De quién se quiere proteger?
- ¿Cuáles son los puntos débiles del sistema?

Después de responder estas preguntas, el administrador del sistema obtendrá un esquema elemental de la situación actual y de las medidas de seguridad básicas que debe tomar.

5.3 Trabajo futuro

Se considera que lo expuesto en esta tesis da la pauta para continuar la investigación en el área del software libre, en especial de las redes de terminales gráficas; a continuación se darán algunas ideas que pueden tomarse como base para trabajos futuros y para mejorar los resultados obtenidos:

- Se propone en primer lugar la mejora de las características del servidor para que éste pueda dar soporte a un mayor número de terminales. De acuerdo a las pruebas que se efectuaron, se pudo observar que la memoria es determinante en el rendimiento de la red, por lo que se recomienda tener 1 GB si se requiere expandir la red de 3 a 10 terminales. También es recomendable utilizar como medio de almacenamiento un disco duro SCSI en lugar de un disco IDE, ya que permite transferencias de datos con una mayor velocidad y de manera más confiable; éste aspecto es importante debido a que en un momento determinado todas las terminales pueden estar utilizando este recurso, por lo que el rendimiento de la red puede disminuir.

- Continuando con la parte de optimización de la red, se propone la implementación del proyecto PXES, que es una alternativa al LTSP, pero a diferencia de éste último permite aprovechar los recursos de las terminales con mejores capacidades, lo que

implica que se puedan correr aplicaciones localmente utilizando su memoria y disco duro, disminuyendo considerablemente la carga al servidor. Cabe mencionar que las terminales con menores capacidades como las 486, pueden seguir con el esquema del LTSP, ya que como se pudo observar en este trabajo, es el óptimo para este tipo de computadoras.

- Una posible aplicación de la red implementada en el laboratorio, es la creación de un cluster; esta arquitectura permite utilizar los recursos de las terminales para ejecutar operaciones de cálculo complejas que requieren mayor tiempo de procesamiento. El software que se propone utilizar es Open Mosix, el cual incorpora al kernel de Linux un potente algoritmo de balanceo de cargas para dividir el trabajo de procesamiento entre todos los equipos. Se debe considerar que las terminales deben poseer características de procesamiento y memoria aceptables para que el tiempo en resolver un problema específico sea mínimo.

- Otro proyecto interesante que se propone es el de la realización de una página Web que permita controlar remotamente dispositivos electrónicos conectados a las terminales, por ejemplo circuitos basados en microcontroladores. Esto permitiría entre otras cosas, monitorear el comportamiento de algunos procesos ligados a estos circuitos los cuales tienen potencialmente muchas aplicaciones. Con esto se pretende ofrecer una infraestructura que permita impulsar a los estudiantes a que desarrollen proyectos de electrónica cada vez más novedosos y funcionales.

- Finalmente, aunque es un proyecto independiente a la institución; con la experiencia adquirida en el desarrollo de esta tesis, se pretende implementar el LTSP en un cybercafe, se propone invertir en un servidor que pueda dar soporte a más de 15 computadoras 486 equipándolas con software de todo tipo, así como de Internet. La renta de los equipos de computo sería realmente mínima tomando en cuenta que pueden proporcionar los mismos resultados que si se utilizaran computadoras Pentium III, con la diferencia de brindar el mismo servicio pero a un menor precio, esto debido al ahorro de licencias, mantenimiento de los equipos, etc., y a su vez las personas se familiarizan con el uso de Linux. Posteriormente se pretende crear un servidor de música para dichas terminales, así como un servidor Samba que permita compartir recursos y archivos con diferentes plataformas de Windows (95, 98, NT, 2000, Me) y de esta manera dichas plataformas puedan acceder a los archivos Linux. El reactivar estas computadoras 486 para darle nuevas aplicaciones, y alargar su ciclo de vida; permitirá tener un negocio rentable con inversiones mínimas.

CONCLUSIONES

La idea básica del proyecto fue rescatar computadoras relativamente obsoletas como las 486 y pentium brindándoles una nueva apariencia de procesamiento gracias a la dependencia hacia un servidor, el cual les brinda sus recursos para lograr obtener unas terminales totalmente funcionales que cuentan con una buena velocidad de procesamiento, acceso a internet, procesadores de texto, software de electrónica, hasta diseños en 3D, etc., y todo bajo un ambiente gráfico totalmente amigable.

Se pudo corroborar todo esto ya que las terminales fueron probadas con éxito en el laboratorio, al principio se experimentó la facilidad de instalación del sistema operativo y una vez que se implementó el LTSP, el mantenimiento así como la actualización del software se efectuaron de una forma sencilla. El sistema realmente demostró su estabilidad y confiabilidad, nos ofrece la libertad de poder ajustarlo a nuestras necesidades, aunado a muchas otras ventajas y beneficios que se encontraron al implementar este proyecto.

También se logró recopilar una gran variedad de software de electrónica totalmente gratuito, el cual ofrece nuevas posibilidades de desarrollo para los alumnos. Además éste tipo de software permite un gran ahorro en licencias, así como de hardware adicional debido a que estas aplicaciones no necesitan de mayores recursos para funcionar adecuadamente.

Además este tipo de terminales bajo Linux no recibe tantos ataques de virus como en Windows, podemos decir que este tipo de sistema es realmente recomendable no sólo para instituciones públicas o pequeñas empresas, sino también para grandes empresas que deseen adquirir un sistema potente, seguro, y muy económico. No requiere de un gasto adicional para capacitación ya que Linux es muy intuitivo por su ambiente gráfico.

Es verdad que en Linux existen ciertos problemas con algunos modems, tarjetas de video e impresoras; donde para obtener una buena configuración se requerirá de varias horas, pero también tenemos la opción de que Linux permite ejecutar muchos programas de Windows mediante emuladores.

Con respecto al tiempo de implementación no hay mayor problema, ya que al hallar la configuración adecuada para una terminal las otras configuraciones son muy similares, esto reduce en gran medida el tiempo de implementación ya que solo es cuestión de configurar ciertas partes.

Al hacer uso de la protección de la memoria entre procesos, de manera que ninguno de ellos detenga el sistema como suele suceder usualmente en Windows, se obtiene una estabilidad determinante para el sistema así mismo cuenta con una mejor

optimización de los recursos de la computadora brindándonos la confiabilidad de realizar varios procesos sin problemas; es un sistema mucho más robusto sobre todo para aplicaciones de alta demanda.

Se optó por utilizar el proyecto LTSP debido a que nos permite aprovechar los recursos de computadoras de baja capacidad, si se contará con computadoras medianamente más potentes se pudó haber utilizado por ejemplo el proyecto de PXES ya que permite aprovechar los recursos de las terminales con mejores capacidades, lo que implica que se puedan correr aplicaciones localmente utilizando su memoria y disco duro, disminuyendo considerablemente la carga al servidor. Es por eso que promovemos el uso de un servidor más potente para lograr atender las peticiones de las terminales ya que éstas cuentan con pocos recursos, y dependen de las características del servidor.

Comprobamos que se eligió la mejor opción para reutilizar a las 486 disponibles, logrando darles nuevamente la funcionalidad que se buscaba así como nuevos campos en los que pueden ser utilizadas estas terminales bajo el concepto del LTSP.

REFERENCIAS

RAYA José Luis, RAYA Cristina, Redes Locales, Ed. Alfaomega Ra-Ma, 2002

FLORIANO Blanco Lino, GALAN Galán Susana, Al día en una hora Linux, Ed. Anaya Multimedia, 1997

CARLING M. DEGLER Stephen DENNIS James, Administración de sistemas Linux, Ed. Prentice Hall

PERRY Hekman Jessica, Linux in a nutshell. A desktop quick reference, Ed. Perry O'reilly, 1997

HUSAIN Kamran, PARKER Tim, Linux unleashed, Ed. Sams published, 1995

CARD Remy, DUMAS Erick, MEVEL Frank, The Linux kernel book, Ed. John Wiler & Suns, 1998

BLANCO J. Vicente, Linux Instalación, administración y uso del sistema, Ed. Ra-ma, 1996

NEGUS Christopher, La biblia de Red Hat Linux 7, Ed. Anaya multimedia, 2001

ESPINOSA H. Juan Carlos, Red Hat Linux 7.0 Instalación y configuración básica, Ed. Alfaomega, 2001

BENTSON Randolph, Inside Linux. A look at operating system development, Ed. Specialized System Consultants, Inc. 1996

LES Fred, DERFLER J. Jr. Frank, Guía de Novel Netware, Ed. Anaya multimedia, 1993

STAN Kelly – Bootle, Como usar Unix sistema V, versión 4.0, Ed. Noriega editores, 1993

MEDIAVILLA Manuel, Seguridad en Unix, Ed. Ra-ma, 1998

WELSH Matt, KAUFMAN Lar, Running Linux, Ed. O'reilly & associates, INC, 1996

STEFAN Strobel, VOLKER Elling, Linux unleashing the workstation in your PC, Ed. Springer, 1997

Página web de la FSF, Sección de Filosofía http://www.fsf.org/philosophy/free-sw.es.html

Página web del Proyecto GNU, Sección de Historia
http://www.gnu.org/gnu/thegnuproject.es.html

Página web del Proyecto GNU, Sección de Licencias
http://www.gnu.org/copyleft/gpl.html

Página web del Proyecto LTSP, Sección de Documentación
http://www.ltsp.org/documentation/ltsp-3.0-4-es.html

Página web de creación de imágenes de arranque http://www.rom-o-matic.net

Página web de Fedora Core http://fedora.redhat.com/download/

Página web del proyecto K12LTSP, Sección de Instalación
http://k12ltsp.org/install.html

Página web acerca de estaciones gráficas diskless
http://www.logiclinux.com/soluciones/diskless/

Página web acerca de experiencias con el LTSP http://www.cignux.org.ar/otkraus.htm

Página web de referencia para implementar un cybercafe
http://webmaster.bankhacker.com/ayuda-linux/consulta/Instalar+Cybercafe/

GLOSARIO

ASCII: (American Standard Code for Information Interchange). Es el código estándar de conjunto de caracteres que cualquier computadora puede entender, usado para representar las letras latinas, en mayúsculas, minúsculas, números, puntuación, etc. Hay 128 códigos estándar ASCII, cada uno de los cuales puede representarse por un número binario de 7 dígitos. Sin embargo, otros conjuntos de caracteres como Latin-1 están comenzando a usarse.

Broadcast: Es un tipo especial de dirección IP usada para enviar un mensaje a todos los nodos de una red.

Cliente: Programa que se usa para contactar y obtener datos de un programa de servidor localizado en otra computadora, generalmente a gran distancia. Cada programa cliente está diseñado para trabajar con uno o más tipos de programas servidores específicos, y cada servidor requiere un tipo especial de cliente.

Demonio: Es un proceso independiente encargado de llevar a cabo una tarea sin la intervención del usuario.

Dirección IP: Dirección que identifica a cada computadora en Internet utilizando una fila de cuatro conjuntos de números separados por puntos.

Diskless: PC o estación de trabajo conectado a una LAN sin disco duro ni disquetes y que depende del almacenamiento en disco de un servidor de archivos conectado a la misma LAN.

DNS: (Domain Name System). Es un sistema que traduce un nombre de dominio a un número IP, y se emplea para facilitar el manejo de direcciones ya que es más fácil recordar un nombre que un grupo de números.

EMACS: (Editing MACroS o Extensible MACro System). Es un editor muy popular para Unix y otros sistemas operativos.

Ethernet: Es la tecnología de red de área local más ampliamente utilizada. Es un estándar de la industria muy extendido, desarrollado originalmente por Xerox y formalizado en 1980 por DEC, Intel y Xerox. Las redes Ethernet transmiten datos a 10/100/1000/ Mbps utilizando un protocolo especificado.

FTP: (File Transfer Protocol). Permite transmitir archivos sobre Internet entre una máquina local y otra remota.

Host: Es una computadora anfitriona que permite a varios usuarios correr aplicaciones o programas diferentes.

Runlevel: Es el estado general de un sistema operativo tipo Unix, el cual es definido por el proceso init.

Script: Es un programa o secuencia de instrucciones que son interpretadas o ejecutadas por otro programa.

Sistema de archivos: Un sistema de archivos es el método con el que la información es introducida en un disco duro. Sistemas operativos distintos generalmente usan sistemas de archivos diferentes.

TCP/IP: (Transmisión Control Protocol/Internet Protocol) familia de protocolos que hacen posible la interconexión y trafico de red en internet.

Telnet: Remote login es un protocolo que forma parte de la suit de protocolos TCP/IP y que permite conectar una terminal remota a su computadora host.

VESA: Es un estándar de modos de video para tarjetas VGA y superiores, que permite programar drivers compatibles con todas las tarjetas graficas que cumplan estas normas, independientemente del chip que incorporen.

CONCLUSIONES

La idea básica del proyecto fue rescatar computadoras relativamente obsoletas como las 486 y pentium brindándoles una nueva apariencia de procesamiento gracias a la dependencia hacia un servidor, el cual les brinda sus recursos para lograr obtener unas terminales totalmente funcionales que cuentan con una buena velocidad de procesamiento, acceso a internet, procesadores de texto, software de electrónica, hasta diseños en 3D, etc., y todo bajo un ambiente gráfico totalmente amigable.

Se pudo corroborar todo esto ya que las terminales fueron probadas con éxito en el laboratorio, al principio se experimentó la facilidad de instalación del sistema operativo y una vez que se implementó el LTSP, el mantenimiento así como la actualización del software se efectuaron de una forma sencilla. El sistema realmente demostró su estabilidad y confiabilidad, nos ofrece la libertad de poder ajustarlo a nuestras necesidades, aunado a muchas otras ventajas y beneficios que se encontraron al implementar este proyecto.

También se logró recopilar una gran variedad de software de electrónica totalmente gratuito, el cual ofrece nuevas posibilidades de desarrollo para los alumnos. Además éste tipo de software permite un gran ahorro en licencias, así como de hardware adicional debido a que estas aplicaciones no necesitan de mayores recursos para funcionar adecuadamente.

Además este tipo de terminales bajo Linux no recibe tantos ataques de virus como en Windows, podemos decir que este tipo de sistema es realmente recomendable no sólo para instituciones públicas o pequeñas empresas, sino también para grandes empresas que deseen adquirir un sistema potente, seguro, y muy económico. No requiere de un gasto adicional para capacitación ya que Linux es muy intuitivo por su ambiente gráfico.

Es verdad que en Linux existen ciertos problemas con algunos modems, tarjetas de video e impresoras; donde para obtener una buena configuración se requerirá de varias horas, pero también tenemos la opción de que Linux permite ejecutar muchos programas de Windows mediante emuladores.

Con respecto al tiempo de implementación no hay mayor problema, ya que al hallar la configuración adecuada para una terminal las otras configuraciones son muy similares, esto reduce en gran medida el tiempo de implementación ya que solo es cuestión de configurar ciertas partes.

Al hacer uso de la protección de la memoria entre procesos, de manera que ninguno de ellos detenga el sistema como suele suceder usualmente en Windows, se obtiene una estabilidad determinante para el sistema así mismo cuenta con una mejor

optimización de los recursos de la computadora brindándonos la confiabilidad de realizar varios procesos sin problemas; es un sistema mucho más robusto sobre todo para aplicaciones de alta demanda.

Se optó por utilizar el proyecto LTSP debido a que nos permite aprovechar los recursos de computadoras de baja capacidad, si se contará con computadoras medianamente más potentes se pudó haber utilizado por ejemplo el proyecto de PXES ya que permite aprovechar los recursos de las terminales con mejores capacidades, lo que implica que se puedan correr aplicaciones localmente utilizando su memoria y disco duro, disminuyendo considerablemente la carga al servidor. Es por eso que promovemos el uso de un servidor más potente para lograr atender las peticiones de las terminales ya que éstas cuentan con pocos recursos, y dependen de las características del servidor.

Comprobamos que se eligió la mejor opción para reutilizar a las 486 disponibles, logrando darles nuevamente la funcionalidad que se buscaba así como nuevos campos en los que pueden ser utilizadas estas terminales bajo el concepto del LTSP.

REFERENCIAS

RAYA José Luis, RAYA Cristina, Redes Locales, Ed. Alfaomega Ra-Ma, 2002

FLORIANO Blanco Lino, GALAN Galán Susana, Al día en una hora Linux, Ed. Anaya Multimedia, 1997

CARLING M. DEGLER Stephen DENNIS James, Administración de sistemas Linux, Ed. Prentice Hall

PERRY Hekman Jessica, Linux in a nutshell. A desktop quick reference, Ed. Perry O'reilly, 1997

HUSAIN Kamran, PARKER Tim, Linux unleashed, Ed. Sams published, 1995

CARD Remy, DUMAS Erick, MEVEL Frank, The Linux kernel book, Ed. John Wiler & Suns, 1998

BLANCO J. Vicente, Linux Instalación, administración y uso del sistema, Ed. Ra-ma, 1996

NEGUS Christopher, La biblia de Red Hat Linux 7, Ed. Anaya multimedia, 2001

ESPINOSA H. Juan Carlos, Red Hat Linux 7.0 Instalación y configuración básica, Ed. Alfaomega, 2001

BENTSON Randolph, Inside Linux. A look at operating system development, Ed. Specialized System Consultants, Inc. 1996

LES Fred, DERFLER J. Jr. Frank, Guía de Novel Netware, Ed. Anaya multimedia, 1993

STAN Kelly – Bootle, Como usar Unix sistema V, versión 4.0, Ed. Noriega editores, 1993

MEDIAVILLA Manuel, Seguridad en Unix, Ed. Ra-ma, 1998

WELSH Matt, KAUFMAN Lar, Running Linux, Ed. O'reilly & associates, INC, 1996

STEFAN Strobel, VOLKER Elling, Linux unleashing the workstation in your PC, Ed. Springer, 1997

Página web de la FSF, Sección de Filosofía http://www.fsf.org/philosophy/free-sw.es.html

Página web del Proyecto GNU, Sección de Historia
http://www.gnu.org/gnu/thegnuproject.es.html

Página web del Proyecto GNU, Sección de Licencias
http://www.gnu.org/copyleft/gpl.html

Página web del Proyecto LTSP, Sección de Documentación
http://www.ltsp.org/documentation/ltsp-3.0-4-es.html

Página web de creación de imágenes de arranque http://www.rom-o-matic.net

Página web de Fedora Core http://fedora.redhat.com/download/

Página web del proyecto K12LTSP, Sección de Instalación
http://k12ltsp.org/install.html

Página web acerca de estaciones gráficas diskless
http://www.logiclinux.com/soluciones/diskless/

Página web acerca de experiencias con el LTSP http://www.cignux.org.ar/otkraus.htm

Página web de referencia para implementar un cybercafe
http://webmaster.bankhacker.com/ayuda-linux/consulta/Instalar+Cybercafe/

GLOSARIO

ASCII: (American Standard Code for Information Interchange). Es el código estándar de conjunto de caracteres que cualquier computadora puede entender, usado para representar las letras latinas, en mayúsculas, minúsculas, números, puntuación, etc. Hay 128 códigos estándar ASCII, cada uno de los cuales puede representarse por un número binario de 7 dígitos. Sin embargo, otros conjuntos de caracteres como Latin-1 están comenzando a usarse.

Broadcast: Es un tipo especial de dirección IP usada para enviar un mensaje a todos los nodos de una red.

Cliente: Programa que se usa para contactar y obtener datos de un programa de servidor localizado en otra computadora, generalmente a gran distancia. Cada programa cliente está diseñado para trabajar con uno o más tipos de programas servidores específicos, y cada servidor requiere un tipo especial de cliente.

Demonio: Es un proceso independiente encargado de llevar a cabo una tarea sin la intervención del usuario.

Dirección IP: Dirección que identifica a cada computadora en Internet utilizando una fila de cuatro conjuntos de números separados por puntos.

Diskless: PC o estación de trabajo conectado a una LAN sin disco duro ni disquetes y que depende del almacenamiento en disco de un servidor de archivos conectado a la misma LAN.

DNS: (Domain Name System). Es un sistema que traduce un nombre de dominio a un número IP, y se emplea para facilitar el manejo de direcciones ya que es más fácil recordar un nombre que un grupo de números.

EMACS: (Editing MACroS o Extensible MACro System). Es un editor muy popular para Unix y otros sistemas operativos.

Ethernet: Es la tecnología de red de área local más ampliamente utilizada. Es un estándar de la industria muy extendido, desarrollado originalmente por Xerox y formalizado en 1980 por DEC, Intel y Xerox. Las redes Ethernet transmiten datos a 10/100/1000/ Mbps utilizando un protocolo especificado.

FTP: (File Transfer Protocol). Permite transmitir archivos sobre Internet entre una máquina local y otra remota.

Host: Es una computadora anfitriona que permite a varios usuarios correr aplicaciones o programas diferentes.

Runlevel: Es el estado general de un sistema operativo tipo Unix, el cual es definido por el proceso init.

Script: Es un programa o secuencia de instrucciones que son interpretadas o ejecutadas por otro programa.

Sistema de archivos: Un sistema de archivos es el método con el que la información es introducida en un disco duro. Sistemas operativos distintos generalmente usan sistemas de archivos diferentes.

TCP/IP: (Transmisión Control Protocol/Internet Protocol) familia de protocolos que hacen posible la interconexión y trafico de red en internet.

Telnet: Remote login es un protocolo que forma parte de la suit de protocolos TCP/IP y que permite conectar una terminal remota a su computadora host.

VESA: Es un estándar de modos de video para tarjetas VGA y superiores, que permite programar drivers compatibles con todas las tarjetas graficas que cumplan estas normas, independientemente del chip que incorporen.

REFERENCIAS

RAYA José Luis, RAYA Cristina, Redes Locales, Ed. Alfaomega Ra-Ma, 2002

FLORIANO Blanco Lino, GALAN Galán Susana, Al día en una hora Linux, Ed. Anaya Multimedia, 1997

CARLING M. DEGLER Stephen DENNIS James, Administración de sistemas Linux, Ed. Prentice Hall

PERRY Hekman Jessica, Linux in a nutshell. A desktop quick reference, Ed. Perry O'reilly, 1997

HUSAIN Kamran, PARKER Tim, Linux unleashed, Ed. Sams published, 1995

CARD Remy, DUMAS Erick, MEVEL Frank, The Linux kernel book, Ed. John Wiler & Suns, 1998

BLANCO J. Vicente, Linux Instalación, administración y uso del sistema, Ed. Ra-ma, 1996

NEGUS Christopher, La biblia de Red Hat Linux 7, Ed. Anaya multimedia, 2001

ESPINOSA H. Juan Carlos, Red Hat Linux 7.0 Instalación y configuración básica, Ed. Alfaomega, 2001

BENTSON Randolph, Inside Linux. A look at operating system development, Ed. Specialized System Consultants, Inc. 1996

LES Fred, DERFLER J. Jr. Frank, Guía de Novel Netware, Ed. Anaya multimedia, 1993

STAN Kelly – Bootle, Como usar Unix sistema V, versión 4.0, Ed. Noriega editores, 1993

MEDIAVILLA Manuel, Seguridad en Unix, Ed. Ra-ma, 1998

WELSH Matt, KAUFMAN Lar, Running Linux, Ed. O'reilly & associates, INC, 1996

STEFAN Strobel, VOLKER Elling, Linux unleashing the workstation in your PC, Ed. Springer, 1997

Página web de la FSF, Sección de Filosofía http://www.fsf.org/philosophy/free-sw.es.html

Página web del Proyecto GNU, Sección de Historia
http://www.gnu.org/gnu/thegnuproject.es.html

Página web del Proyecto GNU, Sección de Licencias
http://www.gnu.org/copyleft/gpl.html

Página web del Proyecto LTSP, Sección de Documentación
http://www.ltsp.org/documentation/ltsp-3.0-4-es.html

Página web de creación de imágenes de arranque http://www.rom-o-matic.net

Página web de Fedora Core http://fedora.redhat.com/download/

Página web del proyecto K12LTSP, Sección de Instalación
http://k12ltsp.org/install.html

Página web acerca de estaciones gráficas diskless
http://www.logiclinux.com/soluciones/diskless/

Página web acerca de experiencias con el LTSP http://www.cignux.org.ar/otkraus.htm

Página web de referencia para implementar un cybercafe
http://webmaster.bankhacker.com/ayuda-linux/consulta/Instalar+Cybercafe/

GLOSARIO

ASCII: (American Standard Code for Information Interchange). Es el código estándar de conjunto de caracteres que cualquier computadora puede entender, usado para representar las letras latinas, en mayúsculas, minúsculas, números, puntuación, etc. Hay 128 códigos estándar ASCII, cada uno de los cuales puede representarse por un número binario de 7 dígitos. Sin embargo, otros conjuntos de caracteres como Latin-1 están comenzando a usarse.

Broadcast: Es un tipo especial de dirección IP usada para enviar un mensaje a todos los nodos de una red.

Cliente: Programa que se usa para contactar y obtener datos de un programa de servidor localizado en otra computadora, generalmente a gran distancia. Cada programa cliente está diseñado para trabajar con uno o más tipos de programas servidores específicos, y cada servidor requiere un tipo especial de cliente.

Demonio: Es un proceso independiente encargado de llevar a cabo una tarea sin la intervención del usuario.

Dirección IP: Dirección que identifica a cada computadora en Internet utilizando una fila de cuatro conjuntos de números separados por puntos.

Diskless: PC o estación de trabajo conectado a una LAN sin disco duro ni disquetes y que depende del almacenamiento en disco de un servidor de archivos conectado a la misma LAN.

DNS: (Domain Name System). Es un sistema que traduce un nombre de dominio a un número IP, y se emplea para facilitar el manejo de direcciones ya que es más fácil recordar un nombre que un grupo de números.

EMACS: (Editing MACroS o Extensible MACro System). Es un editor muy popular para Unix y otros sistemas operativos.

Ethernet: Es la tecnología de red de área local más ampliamente utilizada. Es un estándar de la industria muy extendido, desarrollado originalmente por Xerox y formalizado en 1980 por DEC, Intel y Xerox. Las redes Ethernet transmiten datos a 10/100/1000/ Mbps utilizando un protocolo especificado.

FTP: (File Transfer Protocol). Permite transmitir archivos sobre Internet entre una máquina local y otra remota.

Host: Es una computadora anfitriona que permite a varios usuarios correr aplicaciones o programas diferentes.

Runlevel: Es el estado general de un sistema operativo tipo Unix, el cual es definido por el proceso init.

Script: Es un programa o secuencia de instrucciones que son interpretadas o ejecutadas por otro programa.

Sistema de archivos: Un sistema de archivos es el método con el que la información es introducida en un disco duro. Sistemas operativos distintos generalmente usan sistemas de archivos diferentes.

TCP/IP: (Transmisión Control Protocol/Internet Protocol) familia de protocolos que hacen posible la interconexión y trafico de red en internet.

Telnet: Remote login es un protocolo que forma parte de la suit de protocolos TCP/IP y que permite conectar una terminal remota a su computadora host.

VESA: Es un estándar de modos de video para tarjetas VGA y superiores, que permite programar drivers compatibles con todas las tarjetas graficas que cumplan estas normas, independientemente del chip que incorporen.

GLOSARIO

ASCII: (American Standard Code for Information Interchange). Es el código estándar de conjunto de caracteres que cualquier computadora puede entender, usado para representar las letras latinas, en mayúsculas, minúsculas, números, puntuación, etc. Hay 128 códigos estándar ASCII, cada uno de los cuales puede representarse por un número binario de 7 dígitos. Sin embargo, otros conjuntos de caracteres como Latin-1 están comenzando a usarse.

Broadcast: Es un tipo especial de dirección IP usada para enviar un mensaje a todos los nodos de una red.

Cliente: Programa que se usa para contactar y obtener datos de un programa de servidor localizado en otra computadora, generalmente a gran distancia. Cada programa cliente está diseñado para trabajar con uno o más tipos de programas servidores específicos, y cada servidor requiere un tipo especial de cliente.

Demonio: Es un proceso independiente encargado de llevar a cabo una tarea sin la intervención del usuario.

Dirección IP: Dirección que identifica a cada computadora en Internet utilizando una fila de cuatro conjuntos de números separados por puntos.

Diskless: PC o estación de trabajo conectado a una LAN sin disco duro ni disquetes y que depende del almacenamiento en disco de un servidor de archivos conectado a la misma LAN.

DNS: (Domain Name System). Es un sistema que traduce un nombre de dominio a un número IP, y se emplea para facilitar el manejo de direcciones ya que es más fácil recordar un nombre que un grupo de números.

EMACS: (Editing MACroS o Extensible MACro System). Es un editor muy popular para Unix y otros sistemas operativos.

Ethernet: Es la tecnología de red de área local más ampliamente utilizada. Es un estándar de la industria muy extendido, desarrollado originalmente por Xerox y formalizado en 1980 por DEC, Intel y Xerox. Las redes Ethernet transmiten datos a 10/100/1000/ Mbps utilizando un protocolo especificado.

FTP: (File Transfer Protocol). Permite transmitir archivos sobre Internet entre una máquina local y otra remota.

Host: Es una computadora anfitriona que permite a varios usuarios correr aplicaciones o programas diferentes.

IEEE802.x: Es un conjunto de normas que definen las características físicas de las redes, dictada por el IEEE (the Institute of Electrical and Electronic Engineers). La mayor parte de las redes cableadas cumplen la norma 802.3, especificación para las redes ethernet basadas en CSMA/CD, o la norma 802.5, especificación para las redes token ring.

Kernel: Núcleo del sistema operativo. Es el que se encarga de las labores de más bajo nivel (el nivel más cercano al hardware) tales como gestión de memoria, de entrada/salida de dispositivos, etc.

LAN: (Red de Área Local). Red privada que interconecta diferentes dispositivos como computadoras, impresoras, scanners, etc., dentro de un mismo edificio con el fin de compartir recursos y establecer comunicación entre ellos.

Modeline: Es un mecanismo que permite describir en detalle el modo de vídeo deseado, asignando parámetros como los tiempos de pixelación, frecuencia de refrescado y la resolución de la pantalla.

Motif: Fue el primer GUI (graphical user interface) con comportamiento y apariencia tipo PC para cualquier sistema que corriera X Window.

NIC: (Network Interfase Card). Es una tarjeta de expansión de dispositivo que constituye la conexión física entre un equipo y una red.

Par trenzado: Termino general que designa sistemas de cableado específicamente diseñados para la transmisión de datos y en los cuales los hilos conductores están trenzados.

POSIX: (Portable Operating System for unIX). La interfaz de sistema operativo portátil para UNIX es una compilación de estándares para los sistemas operativos basados en UNIX.

Protocolo: Es un conjunto formal de normas o procedimientos para poder establecer comunicación entre dos dispositivos.

Puerto: Se llama así a un lugar donde la información entra o sale de una computadora o ambas cosas. En Internet, puerto también se refiere a menudo a un número que es parte del URL, apareciendo tras el signo ":", justo después del nombre de dominio, el cual le indica al servidor por que canal de transmisión debe recibir la petición por parte del cliente.

Punto de montaje: El directorio bajo el que un sistema de archivos resulta accesible después de haber sido montado.

Runlevel: Es el estado general de un sistema operativo tipo Unix, el cual es definido por el proceso init.

Script: Es un programa o secuencia de instrucciones que son interpretadas o ejecutadas por otro programa.

Sistema de archivos: Un sistema de archivos es el método con el que la información es introducida en un disco duro. Sistemas operativos distintos generalmente usan sistemas de archivos diferentes.

TCP/IP: (Transmisión Control Protocol/Internet Protocol) familia de protocolos que hacen posible la interconexión y trafico de red en internet.

Telnet: Remote login es un protocolo que forma parte de la suit de protocolos TCP/IP y que permite conectar una terminal remota a su computadora host.

VESA: Es un estándar de modos de video para tarjetas VGA y superiores, que permite programar drivers compatibles con todas las tarjetas graficas que cumplan estas normas, independientemente del chip que incorporen.