UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO

ESCUELA NACIONAL DE ESTUDIOS PROFESIONALES

ARAGON

LINUX COMO OPCION DE UNIX PARA SISTEMAS DE ARCHIVOS Y SERVIDORES DE INTERNET

T E S I S

Que para obtener el Titulo de:

INGENIERO EN COMPUTACION

P r e s e n t a:

CARLOS ENRIQUE LECHUGA TELLO

San Juan de Aragón, Edo. de México, a 18 de Marzo de 1999

DIRECTOR DE TESIS

Ing. Juan Gastaldi Perez

SINODALES

Ing. Raúl Hector León Berber
Ing. Ricardo Gutierrez Orozco
Ing. Hugo Portilla Vázquez
Ing. Gladis E. Fuentes Chavez

GRACIAS A DIOS POR HABERME DADO ESTOS PADRES , PORQUE ESTE LOGRO ES MAS SUYO QUE MIO YA QUE SIN ELLOS HUBIERA SIDO IMPOSIBLE REALIZARLO.

GRACIAS POR SU ESFUERZO, POR SU APOYO Y SOBRE TODO POR SU INFINITO AMOR.

PARA HORTENCIA Y ENRIQUE
MIS PADRES.

OBJETIVO

El objetivo de este trabajo de investigación es el de presentar al sistema operativo llamado Linux como una opción ventajosa y económica de UNIX para montar servidores de archivos y de Internet en computadoras personales. Así como explicar sus características básicas.

INDICE

INTRODUCCION

Las computadoras son hoy en día una de las herramientas mas necesarias para la vida moderna dado el poder de simplificación y realización de tareas que ofrecen.

Según la definición dada por la Real Academia Española una computadora es una maquina analítica de manipulación y procesamiento de información transformada en datos codificados y estructurados.

Para que esto sea posible se hace necesaria la existencia de dos elementos que se complementan y que en conjunto conforman a una computadora: el hardware y el software. Como hardware se debe de conceptualizar al conjunto de componentes físicos (electrónicos, eléctricos y mecánicos) que realizan las operaciones básicas y soportan la información.

El software es el conjunto de instrucciones y datos que describen y permiten la ejecución del trabajo a realizar. A este conjunto de instrucciones también se le denomina como programa.

El software se divide en software de sistema y en software de aplicación.

El maestro Joyanes Aguilar los define como "... los programas específicos que realizan tareas concretas y determinadas como pueden ser: contabilidad, stock, etc. ... (software de aplicación)" y "... el conjunto de programas que mejoran la eficiencia y rendimiento de las computadoras (software de sistema)...".[1]

El software de sistema esta compuesto esencialmente del sistema operativo, lenguajes de programación y editores de texto.

[1] Joyanes Aguilar, programación Basic para microcomputadoras 3ª edición, Mc Graw Hill, Pag 6

Continuando con el Profesor. Joyanes: "El sistema operativo es el conjunto de programas que administran todos los recursos de la computadora: operaciones en la memoria central, almacenamiento y recuperación de programas y datos de discos y cintas en forma de archivos. operaciones de entrada y salida, comunicación con periféricos etc."[2]

Otra definición de sistema operativo dada por el consultor informático Jack Tackett es "Un sistema operativo es un conjunto complejo de códigos informáticos que proporcionan los protocolos de proceso operativo, o leyes de comportamiento...." [3]

Los sistemas operativos pueden ser monousuarios. es decir que solo una persona pueda trabajar en el mediante una computadora, o multiusuarios, cuando varias personas pueden trabajar con el mismo sistema operativo desde diferentes computadoras enlazadas a una computadora central.

Otra clasificación de los sistemas operativos se basa en su capacidad de realizar una (monotarea) o varias tareas (multitarea) a partir de la misma computadora.

En este sentido, Linux es un sistema operativo multiusuario y multitareas.

La palabra multitarea se refiere a la capacidad de ejecutar varios programas al mismo tiempo sin detener la ejecución de cada aplicación. Linux maneja la multitarea prioritaria, es decir que cada programa tiene garantizada la oportunidad de ejecutarse, y se ejecuta hasta que el sistema operativo da prioridad a otro sistema para que se ejecute.

Multiusuario se refiere a la capacidad de Linux para asignar el tiempo de microprocesador simultáneamente a varias aplicaciones.

[2] Joyanes Aguilar, programación Basic para microcomputadoras 3ª edición, Mc Graw Hill, Pag 7
[3] Tackett Jack y David Gunter; Utilizando Linux 2ª Edición, Prentice Hall, Pag. 10

La característica más remarcable de Linux y sus funciones de multiusuario y multitarea es que mas de una persona puede trabajar con la misma versión de una aplicación al mismo tiempo desde esa terminal o desde terminales distintas.

El sistema operativo Linux fue creado a partir y como opción a el sistema UNIX, bajo una base filosófica contraria a los llamados Derechos Reservados. La finalidad de su primer desarrollador, Linus Torvals, fue crear un sistema tan poderoso y con tantas ventajas como UNIX sin estar sujeto a los caprichos de los fabricantes mediante la posibilidad de modificar el sistema operativo y los programas de aplicación (software de aplicación) para poner software a disposición de quien desee accesar a él.

Puede decirse de Linux que es en sí mismo un proyecto incompleto puesto que fiel a su filosofía, cientos o quizá miles de personas en todo el mundo siguen realizando actualizaciones y ampliaciones al sistema y a sus aplicaciones.

Sin embargo, Linux no es software del dominio publico, existe la propiedad sobre los derechos de autor de sus componentes, pero, éste debe estar disponible para todos y debe ser posible la modificación de los programas para adaptarse a necesidades personales, mediante el acceso a los códigos fuente del programa. Este código siempre viene completo en cualquier versión del sistema y sus programas de aplicación.

Al igual que UNIX, Linux ofrece posibilidades de comunicación y de conexión en red muy superiores a otros sistemas operativos. con Linux se tiene acceso a Internet de una manera económica.

El capitulo 1 "Introducción a Linux" ofrece una descripción detallada del sistema Linux. Presenta las funciones de Linux como son Multitarea, Multiusuario y Shells Programables. También muestra una pequeña reseña histórica de este sistema operativo y nos explica lo que es X-Windows.

El capitulo 2 . "Sistemas de red", proporciona un resumen de lo que son estos sistemas. Se explican los conceptos de procesos centralizados y procesos distribuidos y sus diferencias. También se hace una revisión a las topologías de red mas importantes como lo son la de Anillo, Bus y Estrella, y además se explica el modelo cliente/servidor..

El capítulo 3 "Administración de una red" da una descripción de cómo administrar una red Linux. Se explica el protocolo TCP/IP y la configuración de una red usando este tipo de protocolo, también se explica la configuración de Servicio de Nombres de Dominio (DNS)

El capitulo 4 "Administración de los sistemas de archivo" ofrece un resumen de lo que es el sistema de archivos de Linux. Explica como están organizados los archivos y directorios en Linux.

El capitulo 5 "Uso de Internet" Proporciona una descripción a grandes rasgos de lo que es Internet. Este capitulo es solo como referencia.

CAPITULO 1

INTRODUCCION A LINUX

1 INTRODUCCION A LINUX.

1.1. QUE ES LINUX

Linux es un proyecto de sistema operativo iniciado con la finalidad crear una versión de UNIX para computadoras personales basados en Intel. UNIX es probablemente el sistema operativo más versátil y más popular que se puede encontrar actualmente en estaciones de trabajo de la comunidad científica y profesional. "Un sistema operativo puede ser contemplado como una colección organizada de extensiones software del hardware, consistente en rutinas de control que hacen funcionar una computadora y proporcionan un entorno para la ejecución de los programas. Otros programas se apoyan en las facilidades proporcionadas por el sistema operativo para obtener accesos a los recursos del sistema informático, tales como archivos y dispositivos de entrada/salida (E/S)"[1]

Linux empezó como una afición de Linus Torvalds quien creo este sistema con el objetivo de crear un sustituto del sistema operativo Minix, el cual es un sistema operativo parecido a UNIX disponible para computadoras personales. Desde sus inicios, Linux ha incorporado casi toda la biblioteca de utilidades GNU[2], así como el sistema X Windows, el cual es una interfaz gráfica muy usada en equipos UNIX. GNU pretende garantizar la libertad del usuario de compartir y modificar software, A fin de que este software sea gratuito para todos los usuarios.

Linux es un sistema Multiusuario y Multitarea que observa las especificaciones POSIX y que esta diseñado para procesadores Intel 386 y posteriores.

POSIX es una normativa internacional y se refiere a una familia de estándares, que son desarrollados por IEEE (Instituto de Ingenieros Eléctricos y Electrónicos). Estos estándares detallan las normas para la utilización de sistemas operativos y software en distintos equipos. La siguiente lista presenta algunos de los puntos más importantes de estos estándares.

[1] Milan Milenkovic; Sistemas Operativos 2° Edición; Mc Graw Hill; 1994; Pag 3.
[2] GNU es la licencia publica general

POSIX 1. Indica los servicios que debe proporcionar el kernel.

POSIX 2. Describe las utilerias y características que debe proporcionar cualquier shell.

POSIX 3. Describe las facilidades para la revisión de los sistemas POSIX

POSIX 4 Especifica los servicios de computo de tiempo real que puede ofrecer el kernel.

Linux es el único sistema operativo que en la actualidad esta disponible libremente para proporcionar posibilidades multitarea o multiproceso para múltiples usuarios. Muchas aplicaciones para Linux, al igual que el propio código se encuentran disponibles de forma gratuita en Internet. Así pues, se tiene acceso al código fuente para modificar y ampliar el sistema operativo para adaptarlo a las necesidades del usuario.

Linux es distribuido por muchas organizaciones distintas, cada una con un conjunto de programas específicos, aunque todas proporcionan un núcleo central de archivos que constituyen una versión Linux. Entre las distribuciones de Linux, se pueden mencionar las siguientes:

- Slackware
- MCC Interim Linux
- LST
- SLS
- Debian Linux
- Yggdrasil Plug-and-Play Linux CD-ROM and the Linux Bible
- Trans-Ameritech Linux Plus BSD CD-ROM
- The Linux Quarterly CD-ROM
- Caldera
- Redhat

1.2 DESCRIPCION DE LAS FUNCIONES DE LINUX/UNIX

Las ventajas que se derivan de la utilización del sistema operativo UNIX, y por tanto de Linux, parten de la potencia y flexibilidad de estos sistemas. Estas, son el resultado de muchas funciones incorporadas en el sistema. A continuación se describen las principales que son Multitarea, Multiusuario, Shells Programables, Independencia de sistemas Bajo Linux/UNIX, Comunicaciones y Redes; y Portabilidad de Sistemas Abiertos.

1.2.1 MULTITAREA

Un proceso o tarea es un programa en ejecución. Es la unidad más pequeña de trabajo individualmente planificable por un sistema operativo. Un sistema operativo multitarea se encarga de seguir la pista a todos los procesos activos y les asigna recursos del sistema de acuerdo a políticas ideadas para satisfacer objetivos de rendimiento de diseño.

La palabra multitarea describe la capacidad de ejecutar muchos programas al mismo tiempo sin detener la ejecución de cada aplicación.

Los procesos en un sistema operativo multiproceso se caracterizan por:

- Un proceso para empezar su ejecución ha de residir completamente en memoria y tener asignados todos los recursos que necesite.

- Cada proceso esta protegido del resto de los procesos, ningún otro proceso podrá escribir en las zonas de memoria pertenecientes a ese proceso.

- Los procesos pertenecientes a los usuarios se ejecutan en el modo de usuario del procesador (con restricciones de acceso a los recursos), los que pertenecen al sistema se ejecutaran en el modo kernel del procesador (podrán acceder a cualquier recurso).

- Para que un proceso de usuario acceda a los recursos tendrá que hacerlo por medio de llamadas al sistema.

- Cada proceso tendrá una estructura de datos llamada bloque de control de proceso (BCP), donde se almacenara información acerca del proceso, como:

    - Identificación del proceso (PID)
    - Prioridad
    - Estado del proceso (ejecución, preparado, o suspendido).
    - Estado Hardware (registros y banderas del procesador).
    - Información de planificación y estadísticas de uso.
    - Información de gestión de memoria.
    - Estado de E/S (dispositivos asignados, operaciones pendientes)
    - Información de gestión de archivos(archivos abiertos, derechos)
    - Información de mantenimiento.

- Los procesadores se podrán comunicar, sincronizarse y colaborar entre ellos. Estas operaciones se realizan por:

    - Comunicación: memoria compartida e intercambio de mensajes.
    - Sincronización: semáforos
    - Colaboración: por LPC y RPC (llamadas a procedimientos locales y remotos)

  La principal razón de estas operaciones es que, al residir cada proceso en zonas de memoria independientes, se ha de llamar al sistema para compartir los datos entre los procesos.

- Espacio de direcciones lógicas. En este espacio de direcciones reside el proceso (en la parte baja) y las llamadas al sistema (en la parte alta), esto es así para tener un acceso directo a los recursos del sistema. Este espacio de memoria es igual al máximo que el

sistema operativo es capaz de gestionar (en un sistema de 32 bits se llegará hasta 4 GB), y aquí entra en juego la memoria virtual.

- Los procesos se dividen en trozos de igual tamaño, llamados paginas, cuando se carga un proceso lo que se hace es llevarlo a la memoria virtual y asignarle un numero máximo de paginas en memoria real para emplear.

La multitarea prioritaria garantiza la oportunidad a cada programa de ejecutarse, y la ejecuta hasta que el sistema operativo da prioridad a otro programa para que se ejecute.

La multitarea cooperativa permite que los programas se ejecuten hasta que permiten que se ejecuten otros programas o no tienen nada mas que hacer por el momento.

Linux ejecuta la multitarea prioritaria mientras que MS-DOS y Windows 3.1 ejecutan la multitarea cooperativa.

Para comprender mejor la capacidad multitarea de Linux, se tiene que analizar como trabaja el microprocesador. Este solo puede hacer una cosa a la vez, pero es capaz de realizar esas tareas individuales en periodos de tiempo muy cortos. Por ejemplo, los microprocesadores funcionan a velocidades de 25 a 450 MHZ o más. Esto significa que son capaces de transferir de 25 a 260, o más, millones de bits por segundo. Al procesar un conjunto de instrucciones completo, las velocidades son mucho mayores, generalmente nanosegundos (billonésimas de segundo). La mente humana no puede detectar la diferencia entre un lapso tan corto y algo que se produzca simultáneamente. En resumen parece que las tareas se están realizando al mismo tiempo.

Linux y otros sistemas operativos que ejecutan la multitarea prioritaria consiguen el proceso de prioridad supervisando los procesos que esperan para ejecutarse, así como los que se están ejecutando. El sistema programa entonces cada proceso para que disponga de las mismas oportunidades de acceso al microprocesador. El resultado es que las aplicaciones abiertas parecen estar ejecutándose al mismo tiempo (en realidad hay una

demora de billonésimas de segundo entre el momento en que el procesador ejecuta una serie de instrucciones de una aplicación y el momento programado por Linux para volver a dedicar tiempo a dicho proceso). Es esta capacidad de asignar tiempo a las aplicaciones que se están ejecutando lo que destaca a Linux de otros sistemas operativos y entornos disponibles en la actualidad, como, por ejemplo, Windows 3.11 y Windows 95, MS-DOS y versiones comerciales de UNIX.

Para llevar a cabo la multitarea, Linux supervisa una lista, conocida como cola, de tareas a la espera de ser realizadas. Estas pueden incluir tares de usuario, de sistema operativo, de correo, y tareas en segundo plano. Linux planifica espacios de tiempo de sistema para cada una de ellas.

Linux trabaja por un tiempo con una tarea de la cola, la aparta para comenzar con otra y así sucesivamente; después vuelve a la primera tarea y trabaja de nuevo con ella. Linux continua con estos ciclos hasta que finaliza una tarea y la quita de la cola. En este tipo de arreglo, llamado "compartición de tiempo" se comparten los recursos del sistema entre todas las tareas.

1.2.2 MULTIUSUARIO

Al igual que UNIX, Linux es un sistema multiusuario. Los programadores de UNIX anticiparon que diversas personas querrían usar el sistema, y que varias personas desearían tener archivos personales que no quisieran compartir con los demás usuarios. En un sistema multiusuario, los recursos comunes pueden ser directorios o impresoras.

Un sistema multiusuario permite a los usuarios ajustar sus sesiones de trabajo de acuerdo con sus gustos, sin interferir con los demás que comparten la misma maquina. Un sistema multiusuario facilita designar archivos privados, a los que ningún otro puede tener acceso, así como archivos públicos disponibles para todos.

La capacidad de Linux para asignar el tiempo de microprocesador simultáneamente a varias aplicaciones ha derivado en la posibilidad de ofrecer servicio a diversos usuarios a la vez, ejecutando cada uno de ellos una o más aplicaciones. La característica más remarcable de Linux y sus funciones de Multiusuario y Multitarea es que más de una persona puede trabajar con la misma versión de la misma aplicación al mismo tiempo, desde la misma terminal o desde terminales distintas.

1.2.3 SHELLS PROGRAMABLES

El shell en un sistema operativo se encuentra entre el kernel y el usuario. Un shell es un programa que ejecuta otros programas, este se pone en marcha una vez que se conecta al sistema e interpreta sus comandos. En MS-DOS un programa de este tipo usualmente es llamado interprete de comandos. El más conocido es el COMMAND.COM.

La palabra kernel (que significa núcleo), es la parte medular de un sistema operativo. El kernel es el conjunto de software que proporciona las capacidades básicas del sistema operativo. Los servicios y la seguridad de los programas se apoyan en el kernel. La mayoría de los usuarios no tienen contacto directo con este, ya que sus operaciones se realizan de forma "invisible". Sin embargo, la naturaleza del kernel afecta todo lo que se hace con la computadora. Las capacidades del kernel son probablemente el factor más significativo para determinar lo que pueden y no pueden hacer los programas.

En otras palabras, podemos decir que el shell es el interprete entre el usuario y el kernel del sistema operativo. Lee la línea de comandos que se teclea, determina lo que significa y hace lo que sea necesario para ejecutar esos comandos.

El proceso de exploración del shell se denomina análisis; los comandos se descomponen en componentes que se pueden procesar mas fácilmente. Cada componente se interpreta y ejecuta, incluyendo los caracteres especiales que confieren un significado adicional para el shell. Estos caracteres especiales se amplían aun más en sus correspondientes procesos de comandos y se ejecutan.

"Para comprender lo que hace un shell, considere todo lo que hace COMMAND.COM en el sistema operativo MS-DOS cuando se teclea un comando como FORMAT A:

El proceso para ejecutar este comando involucra los siguientes pasos:

1. El shell le pide que dé un comando (Frecuentemente con la notación C >)

2. Después de que se oprime la tecla Enter, el shell analiza la línea para "imaginarse" que se quiere la ejecución del comando FORMAT.

3. El shell del DOS ejecuta el comando FORMAT encontrando el archivo llamado FORMAT.COM y usando el contenido de ese archivo como el programa FORMAT.

4. Después de que FORMAT termina, el shell le pide que teclee un nuevo comando."[3]

Algunos sistemas operativos solamente reconocen un solo shell. En tales sistemas, solo una parte del software tiene la capacidad de ejecutar otros programas y no se puede usar otra cosa. Sin embargo, los sistemas operativos como el DOS y UNIX no lo atan a un solo shell. Ellos permiten que se cree un shell diferente, y que se use ese shell en vez del estándar. Hay varios shells de amplio uso en los sistemas UNIX/Linux, como el Bourne shell, C shell y Kornshell Aunque muchas versiones de UNIX y Linux incluyen mas de un tipo de shell, todos funcionan básicamente del mismo modo. La principal diferencia entre los tres tipos de shell, radica en la sintaxis de la línea de comandos.

El shell programable es lo que hace que UNIX y Linux sean los sistemas operativos más flexibles de los existentes. De hecho, con el conocimiento necesario en programación, se puede hacer un programa que funcione como shell.

[3] James Garoner; Aprendiendo UNIX 2° Edición; Prentice Hall; 1995; Pag. 6

El shell Bourne es el más antiguo y es el shell original de UNIX. Fue escrito por Steve Bourne. De hecho, Linux utiliza una variación del shell Bourne, el Bash, como shell predeterminado. El shell Bourne posee una gran capacidad de programación.

El shell C, fue desarrollado por Bill Joy en la Universidad de Berkeley. Fue desarrollado para reflejar el hecho de que la informática se estaba haciendo más interactiva. La sintaxis del shell C es muy parecida al lenguaje C. Esta es una de las razones por la que los archivos de secuencias de shell escritos para el shell C a menudo no pueden ejecutarse bajo el shell Bourne o Korn. Pero el shell C tiene algunas características deseables no disponibles en el shell Bourne: edición de comandos, histórico y asignación de alias. El shell Korn, posee todas las características del shell C, pero utiliza la sintaxis del shell Bourne.

En sus formatos más sencillos, los shells Bourne y Korn utilizan el signo de pesos **($)** como indicador estándar; el shell C utiliza el signo de porcentaje **(%)** como su indicador. Estos indicadores pueden cambiarse.

El shell Linux predeterminado es el bash. Este shell proporciona varias características optimizadas, como por ejemplo, edición de comandos e histórico de comandos. Para determinar que shell es el que esta utilizando, introduzca el comando

echo $SHELL

El comando **echo** visualiza en la pantalla de la terminal lo que haya escrito tras la palabra **echo**. SHELL es una variable, mantenida por el shell, que conserva el nombre de su shell actual; ***$SHELL*** es el valor de dicha variable.

Para comprobar si el shell C se encuentra disponible, introduzca el comando

csh

Cuando el signo de porcentaje (%) aparece como indicador, el shell C se encuentra disponible y sé esta ejecutando (introduzca ***exit*** para regresar a su shell anterior). Si aparece un mensaje de error, le indicara que el shell C no se encuentra disponible.

El shell que se utilizara como shell de entrada al sistema se encuentra especificado en el archivo de clave de paso; el ultimo archivo en el registro especifica su shell de entrada al sistema. Debe cambiar este campo para poder modificar su shell de entrada al sistema.

Antes de que se aparezca el indicador del shell, Linux configura su entorno de manera predeterminada. El entorno Linux contiene configuraciones y datos que controlan su sesión mientras permanezca conectado al sistema. Al igual que con el resto de las cosas en Linux, se podrá cambiar libremente cualquiera de estos ajustes de configuración de acuerdo con las necesidades del usuario.

La sesión se encuentra dividida en dos componentes. El entorno terminal y el entorno shell.

Debido a que Linux se ejecuta en una computadora personal, la "terminal" es en realidad el monitor y el teclado.

El entorno terminal.

La sesión de entrada al sistema consta de dos programas independientes que se ejecutan en paralelo para proporcionar la apariencia de tener la maquina en exclusiva. Aunque el shell es el programa que recibe las instrucciones y las ejecuta, antes de que el shell vea los comandos, todo lo que se escriba ha de pasar primero a través de un programa relativamente transparente llamado controlador de dispositivos.

El controlador de dispositivos controla la terminal, recibe los caracteres y decide que hacer con ellos, si es que se debe hacer algo, antes de pasarlos al shell para su interpretación. De la misma forma, cada carácter generado por el shell debe pasar a través del controlador de dispositivos antes de ser trasladado a la terminal.

Linux es el único sistema operativo en que, para un programa, todos los dispositivos conectados al sistema se parecen a cualquier otro dispositivo, y todos los dispositivos parecen archivos. Es tarea de los controladores llevar a cabo esta transformación.

Debido a que la terminal se encuentra siempre conectada al sistema, el controlador de dispositivos le permite definir caracteres especiales, llamados caracteres de control, que sirven como marcadores de fin de archivo y fin de línea para el shell. El controlador de dispositivos también le permite definir caracteres de control que envían señales a un proceso de ejecución (como la señal de interrupción que puede, en la mayor parte de los casos. Detener un proceso en ejecución y devolverlo al shell). La figura 1.1 muestra una forma de comportamiento de kernel, shell y controlador de dispositivos de Linux.

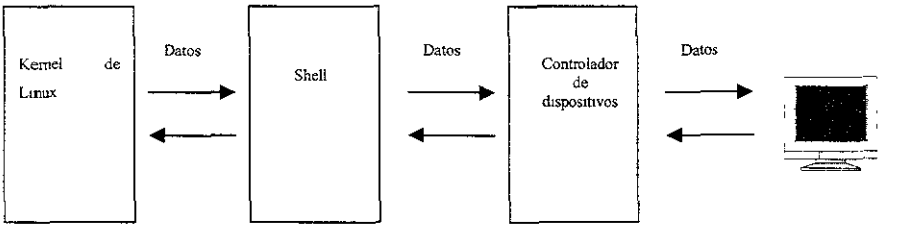

Fig. 1.1 Comportamiento del Kernel, shell y controlado de dispositivos de Linux.

Pueden configurarse varios parámetros para una terminal, aunque la mayor parte de ellos, sean gestionados de forma automática.

El controlador de dispositivos tiene dos modalidades de operación llamadas tratada y sin tratar. En esta, todos los caracteres que se escriben pasan directamente al shell o a un programa ejecutado por el shell. Los programas, como editores y hojas de calculo, necesitan la modalidad sin tratar y la configuran automáticamente. Cuando estos programas terminan, normalmente vuelven a restaurar la terminal a la modalidad tratada. Cuando la terminal se encuentra en la modalidad sin tratar, esta no responde a teclas de control como, por ejemplo, la tecla de interrupción, la de borrar, finalizar, etc.

Cuando la terminal se encuentra en la modalidad tratada, el controlador de dispositivos interpreta cada tecla que se pulsa. Las teclas normales se almacenan en una memoria intermedia hasta que se pulsa la tecla de fin de línea (Enter). Cuando el controlador de dispositivos recibe el carácter de fin de línea, interpreta toda la línea antes de transferir la línea analizada al shell o programa de aplicación.

Entorno shell.

Parte del proceso de conectarse al sistema, es decir, de crear una sesión Linux, es la creación de su entorno. Todos los procesos Linux tienen su propio entorno independiente y distinto del propio programa. Se puede decir que un programa se ejecuta desde dentro de un entorno. El entorno Linux, Conocido como el entorno shell, consiste en un numero de variables y sus valores, que permiten a un programa en ejecución, como un shell, determinar que aspecto tiene el entorno.

Entorno hace referencia a aspectos como el nombre de el shell, el directorio de usuario y que tipo de terminal se esta utilizando. Muchas de estas variables se definen durante el proceso de entrada al sistema y no pueden o no deben modificarse.

1.2.4 INDEPENDENCIA DE DISPOSITIVOS BAJO UNIX.

Probablemente no parezca importante que los periféricos de un sistema puedan funcionar de forma autónoma o independiente. Sin embargo, desde el punto de vista del entorno UNIX multiusuario, se convierte en un factor decisivo. Para comprender la importancia de la independencia de los dispositivos, se debe comprender como contemplan los demás sistemas operativos a los periféricos conectados y como lo hace UNIX.

Hasta hace poco, los sistemas generalmente podían admitir periféricos como impresoras, terminales, unidades de disco y modems. El desarrollo de la tecnología ha ocasionado que el numero de dispositivos crezca de manera elevada. Las dificultades se presentan cuando el usuario no puede utilizar un periférico porque un sistema operativo no puede acceder a

él. Esta incapacidad puede ser resultado de una arquitectura de sistema incompatible, de limitaciones de direccionamiento del sistema operativo, etc.

UNIX evita los problemas que surgen al agregar nuevos dispositivos contemplando cada periférico como un archivo aparte. A medida que se necesitan nuevos dispositivos, el administrador del sistema añade al kernel el enlace necesario. Este enlace, también denominado controlador de dispositivo, garantiza que el kernel y el dispositivo se fusionan del mismo modo cada vez que se solicita el servicio del dispositivo.

A medida que se van desarrollando y facilitando periféricos mejores para el usuario, el sistema operativo UNIX permite un acceso inmediato y sin limitaciones a sus servicios, una vez que se han enlazado los dispositivos con el kernel. La clave de la independencia de los dispositivos estriba en la adaptabilidad del kernel. Otros sistemas operativos solo permiten un determinado numero de un tipo determinado de dispositivos. UNIX puede admitir cualquier cantidad de dispositivos de cualquier tipo, porque cada dispositivo se contempla de forma independiente a través de su enlace individual con el kernel.

Linux comparte muchas de las ventajas de UNIX en cuanto a independencia de los dispositivos se refiere. Aunque Linux puede presentar algunos problemas debido a que como no es comercial, no cuenta se cuenta con una garantía, pero a pesar de esto, se cuenta con una gran cantidad de dispositivos y además se cuenta con un kernel adaptable que se puede programar para que puedan funcionar los nuevos dispositivos.

1.2.5 COMUNICACIONES Y REDES

La superioridad de UNIX sobre otros sistemas operativos es igualmente evidente en sus utilidades de comunicaciones y conexión en red. Linux no es ninguna excepción. Ningún otro sistema operativo incluye unas posibilidades de conexión a red tan ajustadamente acopladas y ningún otro sistema operativo posee la flexibilidad incorporada de estas mismas características. Tanto si se necesita hablar con otro usuario a través de una utilidad

de correo como si se necesita transferir archivos extensos de otro sistema que se encuentra en el otro extremo del país, Linux ofrece los medios para hacerlo.

Linux, fue creado pensando en redes, y su evolución ha sido gracias a ellas. Esto lo hace muy potente, soportando protocolos como TCP/IP, NetBUI, IPX entre otros, permitiendo una conexión directa a Internet, utilizándose , si se desea como servidor de WWW, FTP, gopher, news y todos los demás servicios de Internet.

Además se pueden transferir mensajes internos o archivos mediante varios comandos de Linux.

Linux no solo permite la transferencia de programas y archivos, sino que proporciona a los administradores de sistemas una ventana de acceso a otro sistema. Mediante esta función de acceso remoto, un técnico puede reparar muchos sistemas de forma eficaz, aunque se encuentren a una distancia considerable.

Las posibilidades de comunicación inherentes al sistema operativo se diseñaron para admitir múltiples tareas y múltiples usuarios alejados entre sí. UNIX se ha convertido de forma natural en el sistema favorito dentro del mercado de trabajo, gracias a las mismas características que lo distinguieron en la comunidad científica y educativa, Linux esta definido para seguir el mismo camino como único sustituto de sistemas comerciales UNIX.

1.2.6 PORTABILIDAD DE SISTEMAS ABIERTOS

En los esfuerzos por la estandarización, muchas organizaciones han mostrado interés en la dirección en que se desarrollan los sistemas operativos. UNIX no ha pasado desapercibido. El impulso por la estandarización de UNIX se deriva de sus muchas variaciones disponibles actualmente.

Se han dedicado grandes esfuerzos a combinar y compaginar todas las versiones de UNIX en una única versión del sistema operativo que lo englobe todo. Inicialmente, la tarea

encontró un gran entusiasmo y se invirtieron algunos recursos en llegar a mezclar las distintas versiones, pero esta tarea fracasó debido a que ningún desarrollador quería perder lo invertido en su desarrollo particular.

Sin embargo la existencia de variaciones de UNIX, no representa ningún problema ya que a pesar de las distintas variaciones, todas son esencialmente superiores a los demás sistemas operativos disponibles en la actualidad.

"La portabilidad es simplemente la posibilidad de transportar un sistema operativo de una plataforma a otra sin que se vea alterado su comportamiento. UNIX es decididamente, un sistema operativo portátil. En sus inicios UNIX solo podía funcionar en una plataforma especifica, el miniordenador DEC PDP-7. En la actualidad, las numerosas variantes de UNIX pueden funcionar en cualquier entorno y en cualquier plataforma, desde portátiles hasta mainframes."[4]

La portabilidad ofrece el medio para que varias plataformas informáticas que ejecutan UNIX se comuniquen de forma precisa y eficaz con cualquiera de las demás sin añadir ninguna interfaz de comunicaciones especial, cara o ultimo modelo del mercado. Esto no se puede lograr con ningún otro sistema operativo existente.

1.3 HISTORIA DE LINUX

La historia de Linux esta ligada a la historia de UNIX y a la de un sistema denominado Minix escrita por Andrew Tannebaum. Este era una guía de aprendizaje de sistema operativo. Minix se popularizo en varias plataformas de computadoras personales, incluyendo las computadoras basadas en MS-DOS.

El primer sistema UNIX fue desarrollado por Ken Thomson, que trabajaba en los laboratorios AT&T, a finales de los años sesenta. Fue un sistema de investigación,

[4] Tackett Jack y David Gunter; Utilizando Linux 2° Edición: Prentice Hall, Pag. 22.

construido para probar nuevos conceptos de diseño de sistemas operativos y para proporcionar un ambiente de programación altamente productivo. El sistema operativo UNIX, fue diferente a los sistemas operativos de esos tiempos en varias formas:

- Su propósito fue que solo lo usara un pequeño grupo de gente, a diferencia de los grandes sistemas comerciales que a veces permiten que cientos de personas trabajen simultáneamente en la misma maquina.

- Los programadores supusieron que todos los usuarios de UNIX serian profesionales experimentados.

- UNIX tuvo pocas medidas de seguridad, debido a que todos sus usuarios eran amigos que trabajaban para el mismo grupo de investigación.

- Su propósito fue construir un sistema para el desarrollo de nuevos programas, a diferencia de muchos otros sistemas de aquella época que fueron diseñados para ejecutar a toda hora grandes programas de negocios.

Esos factores hicieron que los programadores se interesaran en UNIX. Poco a poco, UNIX fue ganando auge en las instituciones educativas.

Aunque AT&T fue el creador del sistema operativo UNIX, muchas otras empresas particulares han intentado mejorar el concepto básico a lo largo de los años. A continuación se presentan algunas de las principales variaciones de UNIX que se utilizan en la actualidad.

AT&T

Ken Thompson, un programador de AT&T Bell Laboratories, y un grupo de personas que trabajaban bajo su supervisión desarrollaron un sistema flexible y completamente compatible con las distintas necesidades de los programadores. Este sistema recibió el

nombre de UNIX. Desde ese entonces, UNIX se ha convertido en el estándar para los sistemas operativos multiusuario y multitarea.

BSD

Berkeley Software Distribution (BSD), de la Universidad de California, introdujo la primera versión de UNIX, basándose en la Versión 7 de AT&T, en 1978. A este sistema se le conoce como BSD UNIX. Este sistema contiene algunas mejoras desarrolladas para que UNIX fuera más cómodo para el usuario.

Las mejoras de BSD UNIX, fueron un intento de hacer que UNIX fuera atractivo para los usuarios ocasionales y para los programadores avanzados que apreciaban su flexibilidad para adaptarse a sus exigencias en constante evolución. A pesar de ser menos del 100% de compatible con el UNIX original de AT&T, BSD consiguió sus objetivos: las caracteristicas añadidas atrajeron a los usuarios ocasionales lo suficiente como para utilizar UNIX. BSD se ha convertido en el estándar académico.

USL

UNIX System Laboratories (USL) era una compañía surgida de la organización de AT&T que había desarrollado el sistema operativo UNIX desde principios de la década de los 80. Antes de que Novell la adquiriera en 1993, USL creo el código fuente para todas las variantes del UNIS System V del mercado.

La ultima versión de UNIX por parte de USL fue el UNIX System V versión 4.2 (SVR4.2). El SVR4.2 marco la primera entrada de USL en el mercado de UNIX comercializado. En participación con Novell, que creo temporalmente una compañía denominada Univel, USL produjo una versión comercializable del SVR4.2 denominada UnixWare. Con la adquisición de USL por parte de Novell, esta ha cambiado el enfoque de USL de productor de código fuente a productor de UnixWare.

XENIX, SunOS y AIX

Microsoft desarrollo su versión de UNIX, denominada XENIX, a finales de la década de los 70 y a principios de los 80. El aumento de potencia disponible para las computadoras personales empezó a rivalizar con la de los miniordenadores existentes. Con la aparición del microprocesador 80386 de Intel, pronto se hizo evidente que XENIX, que se había desarrollado específicamente para computadoras personales, ya no era necesario. Microsoft y AT&T fusionaron XENIX y UNIX en un único sistema operativo denominado System V/386 Versión 3.2, que puede funcionar prácticamente en cualquier configuración de hardware.

Sun Microsystems ha contribuido enormemente a la comercialidad de UNIX promocionando el SunOS y sus correspondientes estaciones de trabajo. El trabajo de Sun con UNIX produjo una versión basada en BSD.

La incursión de IBM en el mundo de UNIX dio como resultado un producto denominado AIX (Advanced Interactive Executive). Aunque AIX no es tan conocido como otras versiones de UNIX, su comportamiento es correcto y no causa problemas.

LINUX

Linux es una idea original de un estudiante de informática, llamado Linus Torvalds. Linux, inicio como una afición de Linus, que esperaba crear una versión más sólida de UNIX para usuarios de Minix.

El sistema Minix se escribió para demostrar varios conceptos informáticos que se encuentran en los sistemas operativos. Incorporo estos conceptos en un sistema autónomo que imita a UNIX. El programa estaba disponible para cualquier estudiante de informática del mundo y pronto tuvo un gran grupo de usuarios. Torvalds decidió proporcionar una plataforma mejor para los usuarios de Minix que pudiera ejecutarse en cualquier computadora personal, y se baso en las computadoras con procesador 386 recién

aparecidas, debido a las propiedades de conmutación de tareas de la interfaz en modo protegido del 80386.

1.4 DESCRIPCION GENERAL DE LAS FUNCIONES

1.4.1 FUNCIONES BASICAS

Debido a que Linux es un clon de UNIX, significa que con Linux se obtienen muchas de las ventajas de UNIX. La multitarea de Linux es completamente prioritaria, es decir puede ejecutar varios programas al mismo tiempo y cada programa sigue funcionando, por ejemplo. le permite iniciar una transferencia de archivos, imprimir un documento, copiar un disco flexible y reproducir un CD al mismo tiempo. Otros programas, como, por ejemplo, Microsoft Windows 3.1, permiten ejecutar varios programas, pero cuando pasa de un programa a otro, el primero deja de ejecutarse. Windows 95 y Windows NT se parecen mas a Linux.

Linux es completamente multiusuario, lo que significa que más de una persona puede entrar en el sistema y utilizarlo. Aunque puede que esta característica no sea muy útil en casa, en un entorno de empresa o universitario permite que muchas personas accedan a los mismos recursos al mismo tiempo, sin duplicar la inversión en maquinas costosas. Incluso desde casa, será muy útil la posibilidad de entrar en cuentas separadas a lo que se denominan terminales virtuales.

1.4.2 VENTAJAS DEL USO DE LINUX

Son muchas las ventajas derivadas del uso de Linux. Para las computadoras personales, Linux ofrece un sistema completo con posibilidades multiusuario y multitarea incorporadas que se benefician de toda la potencia de procesamiento de los sistemas informáticos 386 y superiores. Linux incluye una implementación completa del protocolo de red TCP/IP. Con Linux puede conectarse a Internet y disponer de toda la cantidad de información que

contiene; dispone de un sistema e-mail completo para enviar y recibir mensajes por el Internet y también de una completa interfaz gráfica de usuario (GUI) para Linux denominada XFree86, basada en el popular sistema X Windows y que es una implementación completa del sistema X Windows que se puede distribuir libremente con Linux. XFree86 proporciona los elementos habituales que puede encontrar en otras plataformas GUI comerciales como por ejemplo, Windows y OS/2.

Linux permite ampliar la memoria física de RAM empleando un espacio en el disco duro denominado SWAP. Aquí, él puede escribir datos y moverlos del disco duro a la RAM y viceversa, convirtiendo este espacio en parte de la RAM. De esta forma, la memoria se aumentara denominándose memoria virtual.

Linux es el único sistema operativo competitivo que es distribuido con sus códigos fuente. Es decir, que el usuario puede adaptar el sistema o cualquier aplicación para que utilice exactamente el equipo que posea y saque provecho de la configuración que el usuario tenga.

Varios sistemas operativos, requieren que los programas sean autosuficientes, lo que hace que estos programas sean mas grandes de lo que necesitan ser. Los sistemas operativos como UNIX, pueden generar programas que accesan a librerías de código que ya existen en el sistema; de esta manera, el tamaño de los programas se mantiene bajo.

Muchos sistemas operativos guardan todos los procesos en la memoria física hasta que se agota el espacio libre y comienzan a utilizar su memoria virtual. Esto significa que si el usuario deja un proceso abierto, pero no lo usa por un tiempo determinado, este proceso estará usando la memoria, ocasionando que los procesos nuevos tengan que usar el espacio de intercambio.

En los sistemas que manejan la demanda de carga como lo son UNIX y Linux, esto funciona de manera diferente. Cuando los procesos permanecen activos, pero no son

usados, el sistema los manda a su espacio de intercambio, de esta manera se libera la memoria física y podrá ser usada por los procesos que se estén usando.

Aplicaciones

Linux dispone de miles de aplicaciones entre las cuales se incluyen programas para hojas de calculo. manejadores de bases de datos, procesadores de texto, desarrollo de aplicaciones en varios lenguajes y paquetes de comunicaciones para conectarse a redes.

Para los programadores Linux ofrece muchas herramientas para el desarrollo de programas. Dispone de compiladores para muchos de los principales lenguajes de programación actuales, como C, C++ y SmallTalk. Linux proporciona las herramientas para crear los programas propios, Para esto tiene las herramientas Flex y Bison.

Linux también ofrece la posibilidad de comunicarse con los sistemas de la oficina de su empresa. Y si el usuario es un administrador de sistema UNIX, Linux puede ayudar a realizar sus tareas desde casa. Aunque el sistema de trabajo en casa todavía no es muy potente.

Dos de las palabras claves de la industria son sistemas abierto e interoperabilidad. Estos términos se refieren a la posibilidad de que muchos sistemas puedan comunicarse entre si. Muchas especificaciones de sistemas abiertos requieren el cumplimiento de POSIX. Linux cumple con esos estándares en la actualidad. De hecho, Linux se diseño para ofrecer la portabilidad del código fuente, por lo que si se tiene un programa ejecutándose sobre una versión de UNIX, se podrá trasladar de forma relativamente fácil ese sistema a un sistema que ejecute Linux. Las corporaciones insisten en ese tipo de sistemas abiertos para no depender de un solo proveedor. Con UNIX/Linux y los sistemas abiertos, sin embargo, se tiene el control de sus sistemas. Si el sistema operativo no tiene una característica que se necesita, se pueden efectuar los cambios necesarios. Especialmente teniendo en cuenta que, por lo menos con Linux, dispone del código fuente del sistema operativo.

Linux será de gran utilidad para la gente que quiera aprender mas acerca de UNIX. Esta es una versión funcional de UNIX a la que tiene acceso gratuito e ilimitado. La mayoría de usuarios de UNIX obtienen cuentas en maquinas UNIX que solamente les otorgan determinados derechos y privilegios. Hay algunos comandos de UNIX/Linux que un usuario normal no puede utilizar ni experimentar con ellos, con lo cual es imposible aprender todos los comandos de UNIX. Sin embargo, con Linux se puede hacer lo que el usuario desee.

1.4.3 DESVENTAJAS DE LINUX

Quizás la mayor desventaja de Linux es el hecho de que ninguna entidad corporativa se encarga de su desarrollo.

Es indudable que el hecho de que Linux no disponga de asistencia técnica puede ser un problema. Lo mismo sucede con las aplicaciones para Linux; pues, aunque hay algunos programas comerciales para Linux, la mayoría están desarrollados por pequeñas comunidades que posteriormente se ponen a disposición de todos.

Otra desventaja es que Linux puede resultar difícil de instalar y no funciona en todas las plataformas de hardware. Al contrario que la operación de desarrollo de un programa comercial, donde un equipo compacto de desarrollados pasa bastante tiempo construyendo y probando un programa bajo distintas condiciones y con distinto hardware, los desarrollados de Linux se encuentran repartidos por todo el mundo. No hay ningún programa formal de garantía de calidad. Los desarrollados lanzan sus programas cuando les parece. Además, el hardware admitido por Linux depende del hardware de que disponga cada desarrollador en el momento de escribir esa parte del código.

A continuación se describen los requerimientos de hardware:

El CPU del sistema debe ser compatible con Intel 80386 o posterior, tales como 486 o Pentium. Otros CPU clonicos, como, por ejemplo, Los chips fabricados por Cyrix y

Advanced Micro Devices (AMD) también son compatibles con Linux. Linux no requiere de un coprocesador matemático, pero reduciría notablemente la velocidad del sistema.

Linux solo trabaja con bus tipo ISA y EISA, con el bus local VESA y el bus PCI. No admite la Arquitectura Microcanal (MCA).

Linux requiere de al menos 2 MB de memoria RAM para ejecutarse aunque es recomendable tener 4 MB. Pero si se tiene menos de 4 MB de RAM, deberá de utilizar un archivo de intercambio.

Para mejorar el rendimiento de Linux, se debe de tener un disco duro con un controlador compatible con IBM AT. Linux admite todos los controladores MFM e IDE así como la mayor parte de los controladores RLL y ESDI. Linux admite varios controladores SCSI.

El espacio que Linux necesita en disco es de al menos 20 MB, aunque esto también depende del software que se desee instalar. Aunque lo recomendable es de 150 a 200 MB para una instalación completa.

Otra desventaja es que las aplicaciones para los sistemas operativos como DOS y OS/2 probablemente no funcionaran bajo Linux.

1.5 EL SISTEMA X WINDOWS

El sistema X Windows es un entorno operativo gráfico que da soporte a muchas aplicaciones en la red.

El sistema X Windows surgió de un esfuerzo cooperativo entre dos secciones del MIT; la responsable de un programa de red denominado Athena Project y una sección llamada Laboratorio de ciencia informática, las dos secciones se pusieron de acuerdo para crear una interfaz gráfica de usuario (GUI) para estaciones de trabajo UNIX. A el sistema creado de esta unión se le llamo X Windows.

Después varios programadores formaron una organización que se llamo X Consortium, para promocionar y estandarizar X Windows.

XFree86 es una versión de X Windows que viene con Linux y es una marca registrada de XFree86 Project, Inc. Los mismos programadores que llevaron el X Windows a la plataforma 80386 decidieron comenzar el proyecto para hacerse miembros del X Consortium

X Windows es una serie de piezas que trabajan conjuntamente para presentar al usuario una GUI. El sistema base de ventana es un programa que proporciona servicio al sistema X Windows. La pieza siguiente es un programa para comunicación en la red: el protocolo X Network. Por encima del protocolo de implementación de X Network, esta una interfaz de bajo nivel, que se encuentra entre el sistema base de red y los programas de mas alto nivel. Esta interfaz de bajo nivel se llama Xlib. Los programas de aplicación normalmente utilizan funciones Xlib en vez de otras aplicaciones de bajo nivel. Un administrador de pantallas es el que une esas piezas en un conjunto. El administrador de pantallas es una aplicación X Windows cuyo propósito es controlar como se presentan las pantallas a los usuarios.

El sistema base de ventanas no proporciona los objetos de interfaz de usuario, como barra de desplazamiento, botones de comandos o menús. En esto se diferencia de la mayoría de los demás sistemas de ventanas. Los elementos de interfaz de usuario se dejan para los componentes de capas más altas y para el administrador de pantallas.

X Windows implementa un administrador de ventanas para realizar la tarea de crear y controlar la interfaz que compone la porción visual de sistema.

XFree86 incluye bibliotecas de programas y archivos para programadores que quieran desarrollar sus propias aplicaciones.

X Windows es un sistema cliente/servidor controlado por dos piezas de software, una ejecutándose en el cliente y otra en el servidor. Las piezas cliente y servidor pueden estar

en sistemas distintos o, como en el caso de la mayoría de las computadoras personales, ambos pueden residir en la misma maquina.

El sistema básico de ventanas proporciona a X Windows una gran cantidad de operaciones gráficas de mapas de bits. X Windows y las aplicaciones de X Windows utilizan esas operaciones para presentar a los usuarios información en forma gráfica. XFree86 ofrece solapamiento de ventanas, dibujo inmediato de gráficos, imágenes y gráficos de alta resolución de mapas de bits y texto de alta calidad. Mientras que los sistemas iniciales de X Windows eran en su mayoría monocromáticos, ahora X Windows y XFree86 admiten una amplia gama de sistemas en color.

X Windows también admite las posibilidades de multiproceso de UNIX; de este modo, XFree86 admite las posibilidades de multiproceso de Linux. Cada ventana que se muestra bajo X Windows puede ser una tarea distinta que se ejecuta bajo Linux.

El X Consurtium quería hacer de X Windows un estándar en las estaciones de trabajo UNIX, y esta es una de las razones por las que X Windows se distribuye libremente por Internet. Esta distribución gratuita de X Windows promueve la inter-operabilidad, que es la parte principal de los sistemas abiertos.

Los sistemas que se ejecutan en X Windows suelen tener algún dispositivo para señalar, generalmente un ratón. XFree86 necesita un ratón o algún otro dispositivo parecido. X Windows convierte en eventos las señales que recibe tanto del Mouse como del teclado y a continuación responde a esos eventos realizando las acciones apropiadas.

CAPITULO 2

SISTEMAS DE RED

2. SISTEMAS DE RED

"Una red es un sistema de interconexión entre computadoras. Para ello, es necesario contar, además de con las computadoras correspondientes, con las tarjetas de red y los cables de conexión entre ellas."[5]

Si se conectan todas las computadoras dentro de una misma área, se denomina LAN (Local Area Network) y si se encuentran instaladas en lugares diferentes, WAN (Wide Area Network).

Para la construcción de una red, se debe de contar con una topología, mas adelante en este mismo capitulo se explicaran los principales tipos de topologías.

En los sistemas de red existen diferentes protocolos, que son las reglas y procedimientos que han de seguirse para realizar actividades sobre una red. Entre estas se encuentran IPX, TCP/IP, NETBIOS, XNS.

Entre las ventajas de utilizar una red de computadoras, podemos mencionar las siguientes:

- Posibilidad de compartir impresoras, módem, fax, etc.
- Posibilidad de compartir programas, bases de datos, etc.
- Remplazar o complementar computadoras.
- Establecer enlaces con mainframes.

Un servidor es una computadora de gran potencia, que hace que los recursos disponibles estén accesibles para cada una de las computadoras conectadas en la red.

[5] José Luis Raya, Novel Netware 4, Addison Wesley, P 1.

Los servidores pueden ser de tres tipos: de archivos, de impresión, y de comunicaciones.

Un servidor de archivos mantiene los archivos en subdirectorios privados y compartidos para los usuarios de la red. Estos servidores, pueden ser dedicados o no dedicados, según se dedique solo a la gestión de la red o, además, se puedan utilizar como estación de trabajo. La conveniencia de usar uno u otro va a estar indicada por el numero de estaciones de trabajo de que se vaya a disponer; cuanto mayor sea el numero de ellas, mas conveniente será disponer de un servidor dedicado.

Un servidor de impresión tiene conectadas una o mas impresoras que comparte con los demás usuarios.

Un servidor de comunicaciones permite enlazar diferentes redes locales.

2.1 COMPRENSIÓN DE LOS CONCEPTOS MULTIUSUARIO.

Un sistema multiusuario utiliza dos conceptos principales: servicios de multitarea y multiusuario. Linux tiene la posibilidad de ejecutar varias tareas de modo concurrente-transparente al usuario.

Cada tarea, bien sea un comando sencillo introducido en la línea de comandos o una aplicación compleja, inicia uno o varios procesos. Todo lo que se ejecuta en un sistema Linux esta asociado con un proceso; Linux es un sistema multitarea debido a que ejecuta varios procesos al mismo tiempo.

Hay varias formas de conectarse a un servidor. Se puede utilizar una terminal o una computadora que puede estar ubicado físicamente cerca del servidor, conectado con un cable o al otro lado del planeta conectado por medio de líneas de datos de alta velocidad o por medio de una línea de teléfono normal.

La manera en la que se utilice una terminal, y la manera en que este conectada al servidor, determinan si los recursos de la computadora se consideran distribuidos o centralizados.

Un sistema operativo de un solo usuario, como lo es el MS-DOS, esta diseñado para ser usado por una persona a la vez. Todo el proceso se realiza en una computadora que tiene el único acceso a los recursos, como, por ejemplo, impresoras, almacenamiento y proceso. Los sistemas multiusuario utilizan los modelos del proceso centralizado y distribuido para dar servicio a muchos usuarios al mismo tiempo.

En un entorno de proceso centralizado muchos usuarios acceden a los recursos de una computadora: almacenamiento, impresión, memoria y procesamiento.

En un entorno de proceso distribuido, el procesamiento puede realizarse en la estación de trabajo del usuario y el procesador central se utiliza para distribuir aplicaciones y datos. Las impresoras y los sistemas de almacenamiento pueden estar conectados a la estación de trabajo del usuario o en el servidor principal.

2.2 SISTEMAS DE PROCESO CENTRALIZADO

Debido al avance tecnológico, los sistemas operativos comenzaron a permitir que varios usuarios compartieran recursos de terminales distintas. Dos usuarios podían, en una secuencia de proceso por lotes, ejecutar dos conjuntos de instrucciones al mismo tiempo que compartían un procesador, el almacenamiento y la salida.

Con la llegada de la red telefónica conmutada, las computadoras comenzaron a utilizar los recursos telefónicos para extender geográficamente los recursos informáticos. Con este modelo, cada procesador utilizaba recursos de proceso de comunicaciones para conectarse con terminales remotas. Esto hizo necesario que las terminales y computadoras se pudieran comunicar de una forma mejor. El resultado fue el desarrollo del procesamiento de interfaz de cara al usuario para las tareas de comunicación y el modelo de procesamiento centralizado.

La mayoría de los sistemas UNIX utilizaban el modelo de proceso centralizado. Con este tipo de modelo los mainframe se encargaban de todo el procesamiento. Los usuarios conectados al

mainframe comparten sus recursos. Este modelo cada vez se utilizan menos hoy en día, aunque es adecuado para las ubicaciones informáticas cuyos usuarios están dispersos geográficamente.

Por ejemplo, un banco puede tener un centro principal de procesamiento y todas las sucursales pueden acceder al centro de datos independientemente de su ubicación. Cada usuario cuenta con una comunicación directa con el mainframe de modo que pueda acceder a los recursos centralizados: procesamiento, impresión y almacenamiento (ver figura 2.1).

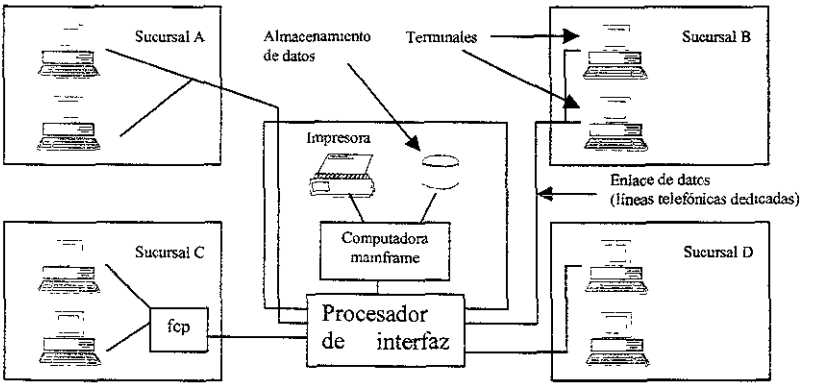

Fig 2.1 Esta figura muestra el modelo de proceso centralizado en un entorno informático.

Según un usuario solicita datos, esta petición se procesa en la computadora de la oficina principal del banco. Los resultados del procesamiento se devuelven a la terminal de la sucursal. Todos los datos son procesados y almacenados por el mainframe.

2.2.1 ELEMENTOS DEL MODELO DE PROCESAMIENTO CENTRALIZADO

Para hacer que funcione un modelo de procesamiento centralizado se necesitan muchos elementos, como el servidor, procesadores de interfaz de cara al usuario, terminales, modems y adaptadores multipuerto.

Servidor: Un servidor es una computadora configurada para compartir sus recursos (impresoras, espacio de almacenamiento, procesos, etc.)

Procesador de interfaz de cara al usuario: Un procesador de interfaz de cara al usuario conecta los canales de comunicación y el servidor. Se encarga de los detalles de la comunicación de modo que el servidor este libre para procesar los datos.

Terminales: Existen dos tipos de terminales, las terminales tontas y las terminales inteligentes. Tradicionalmente, UNIX se utilizaba con terminales tontas que solo cuentan con el teclado y el monitor. Lo mas importante a tomar en cuenta para las terminales tontas, es que no tienen potencia local de proceso. El puerto de comunicaciones de la terminal esta conectada al servidor a través de un módem o directamente. Cuando se escribe en la terminal, las pulsaciones son transportada al servidor en donde son procesadas.

Las terminales inteligentes pueden realizar procesos mínimos localmente. Las cajas registradoras son un ejemplo de terminales inteligentes. El dispositivo local almacena la petición de transacción y la envía completa en vez de transmitir cada pulsación, como lo hace una terminal tonta.

Módem: Para atender las necesidades crecientes de comunicación con otras computadoras se fabricaron unos dispositivos encargados de convertir las señales acústicas que se introducen en ellas en códigos binarios, códigos que son aptos para ser usados por el microprocesador. A estos dispositivos se les conoce como modems (abreviatura de MOdulación DEModulación).

La conexión entre ordenadores se realiza aprovechando la infraestructura ya establecida en todo el mundo: la red telefónica. La línea telefónica se diseño para la transmisión de la voz humana, y no es capaz de transmitir los dos posibles valores de voltaje o código binario. La misión del módem consiste en transformar las señales binarias de una computadora en sonidos dentro del rango de frecuencias de la voz humana (entre 300 y 3400 Hz), es decir, la modulación de dichas frecuencias para convertir la información digital de la computadora en señales analógicas (convertir el código binario en frecuencias acústicas). A este proceso se le conoce como modulación, siendo la demodulacion el procedimiento inverso.

El baudio es una medida de señalización eléctrica, que indica cuantos impulsos se envían por segundo a una red eléctrica.

Los bits por segundo (bps) son una medida de información que indica la velocidad de transferencia de bits, es decir, cuantos bits de información son enviados cada segundo.

Los baudios y los bits por segundo solo coinciden cuando el módem envía un bit por cada baudio o impulso. Sin embargo, empleando diferentes sistemas eléctricos de modulación, es posible enviar mas de un bit por cada baudio.

2.3 SISTEMAS DE PROCESAMIENTO DISTRIBUIDO

Un sistema distribuido consiste en una colección de computadoras autónomas enlazadas mediante una red de computadoras y con software para sistemas distribuidos, este software permite a las computadoras coordinar sus actividades y compartir los recursos del sistema, hardware, software e información.

La potencialidad de los sistemas distribuidos, se volvió aparente a principios de los años 70, años después del surgimiento de las minicomputadoras y de las redes de área local (LANS). El uso de las minicomputadoras como estaciones de trabajo monousuario y su efectividad para el desarrollo de software y otras tareas interactivas fueron influencias importantes. Pero continuaban ausentes, el hardware y software necesarios para hacer que estas computadoras pudieran ser utilizadas cooperativamente.

Entre el periodo de 1971-1980 un equipo investigación de la compañía Xerox llamada Xerix PARC desarrollo, las primeras estaciones de trabajo, servidores de archivos, la primera red local de alta velocidad (Ethernet) y varios sistemas distribuidos experimentales. La primera estación desarrollada por Xerox PARC fue ALTO. Las estaciones de trabajo ALTO, estaban enlazadas por Ethernet. Estas, fueron utilizadas tanto como computadoras personales como plataformas de servidor.

Los sistemas distribuidos necesitan de software completamente distinto al de los sistemas centralizados.

En el procesamiento distribuido, la terminal se sustituye por una estación de trabajo, que en si misma es una computadora, normalmente ejecutando DOS o UNIX. Los programas pueden estar situados y ejecutarse desde el servidor o desde la estación de trabajo. Del mismo modo, los archivos pueden estar situados en cualquiera de los dos sistemas. Si se procesa un archivo en la estación de trabajo, luego se almacena en el servidor de modo que otros pueden acceder a este archivo. Se puede imprimir en impresoras conectadas a la estación de trabajos o en impresoras conectadas al servidor.

La figura 2.2 muestra el ejemplo del banco visto en la sección anterior, pero usando un sistema de procesamiento distribuido.

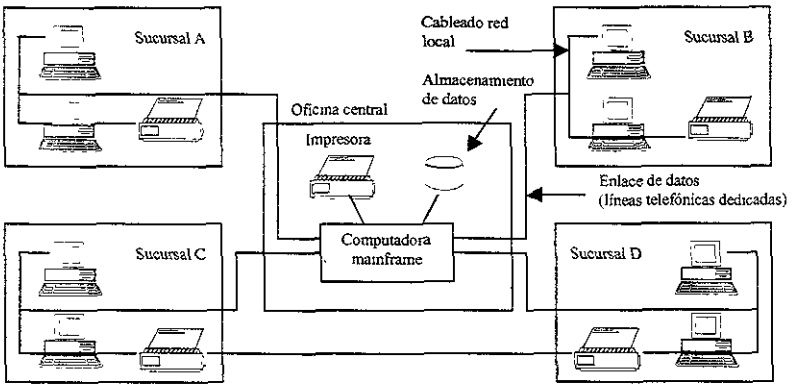

Fig. 2.2 Esta figura muestra el modelo de procesamiento distribuido de un sistema informático.

Para el buen funcionamiento de un sistema distribuido, se deben de contemplar 6 características principales. Estas son: Compartición de recursos, Apertura, Concurrencia, Escalabilidad, Tolerancia a fallos, y Transparencia.

Compartición de recursos. Los beneficios del acceso compartido a un mismo archivo que tenga una base de datos, programas, documentación y cualquier otra información, fueron conocidos, con el surgimiento de sistemas multiusuario o sistemas de tiempo compartido a principios de los 60, y en sistemas multiusuario UNIX a principios de los 70.

- Dispositivos como impresoras, discos y otros periféricos, son compartidos para reducir costos.

- EL compartir información es un requerimiento esencial en muchas aplicaciones.

  - Desarrolladores de software trabajando en equipo, necesitan acceder al trabajo de cada uno de los desarrolladores y pueden compartir la misma herramienta de desarrollo, lo cual permite usar solo una copia de los compiladores, librerías de procedimientos, editores, etc.; siempre que se instale una nueva herramienta de desarrollo, todos los usuarios tendrán acceso inmediato a esta.

Los recursos de una computadora multiusuario están normalmente compartidos entre todos los usuarios, pero los usuarios de una estación de trabajo de un solo usuario o de una computadora personal, no obtienen automáticamente los beneficios de la compartición de los recursos. Los recursos en un sistema distribuido están físicamente encapsulados dentro de una de las computadoras y solamente pueden ser accesadas por otras computadoras. Para una compartición efectiva, cada recurso debe ser administrado por un programa que ofrece una interfaz de comunicación habilitando que el recurso sea accesado, manipulado y actualizado.

El termino administración de recursos es usada para hacer referencia a un modulo de software que administra un conjunto de recursos de una clase en particular. Cada clase de registro requiere de algunos métodos de administración específicos, aunque también de requerimientos comunes. Estos incluyen, El dotar de nombre a cada clase de recurso, habilitar recursos individuales para ser accesados desde cualquier locación, el mapeo de nombres de recursos a direcciones de comunicación y la coordinación de accesos concurrentes que cambian el estado de recursos compartidos para asegurar su consistencia.

Apertura. La apertura de un sistema distribuido es la característica que determina si el sistema puede ser extendido en diferentes formas. Un sistema puede ser abierto o cerrado con respecto a extensiones de hardware. Por ejemplo, la adición de periféricos, memoria o interfaces de comunicación, o con respecto a extensiones de software, la adición de sistemas operativos, protocolos de comunicación y servicios para compartir recursos. La apertura de sistemas distribuidos esta determinada por el grado en que nuevos recursos compartidos pueden ser adheridos sin causar conflicto en el sistema.

La apertura es conseguida especificando y documentando interfaces de software de un sistema y haciendo que puedan ser accesados por otros desarrolladores de software.

Históricamente, los sistemas estaban completamente cerrados. No se les permitía a los programadores extender la semántica de los lenguajes.

UNIX Fue un primer ejemplo del diseño de sistemas abiertos. Este incluye un lenguaje de programación C, lo que permite a los desarrolladores accesar a todos los recursos administrados por el sistema operativos.

Los recursos del sistema operativo UNIX son accesadas a través de un conjunto de procedimientos (denominadas llamadas del sistema) que están completamente documentadas y a disposición de programas escritos en C o cualquier otro lenguaje de programación que soporte las facilidades de las llamadas convencionales del microprocesador. Cuando se instala un nuevo periférico en un sistema UNIX el sistema operativo puede ser extendido para que los programas de aplicación puedan accesar a este, añadiendo nuevas llamadas al sistema.

UNIX es uno de los sistemas operativos mas abiertos debido a:

- Los desarrolladores de aplicaciones tienen acceso a un alto rango de facilidades que ofrece el sistema.

- Para administradores de sistemas, debido a que el sistema operativo puede ser extendido de manera relativamente fácil para agregar nuevos periféricos o controladores de red.

- Para los usuarios de software debido a que es independiente del hardware, Los desarrolladores de software pueden producir programas que pueden funcionar en cualquier computadora sin importar el fabricante.

Concurrencia Con concurrencia nos referimos a la acción de que varios procesos se ejecutan en una misma computadora. Si la computadora esta equipada con un solo procesador central, esto se lleva a cabo intercalando la ejecución de los procesos. Si la computadora tiene N procesadores, entonces N procesos pueden ejecutarse al mismo tiempo, es decir en paralelo.

Los sistemas distribuidos cuentan con varias computadoras, cada una de las cuales tiene uno o mas procesadores. Si hay N computadoras en un sistema distribuido con un solo procesador, entonces, N procesos pueden ejecutarse en paralelo, tomando en cuenta que cada proceso se encuentra en una computadora diferente.

Escalabilidad. Los sistemas distribuidos, operan efectiva y eficientemente en diferentes tamaños de redes. El mas pequeño sistema distribuido, probablemente consiste en dos estaciones de trabajo y en un servidor de archivos, mientras que un sistema distribuido construido alrededor de una red de área local puede contener varios cientos de estaciones de trabajo y muchos servidores de archivos, servidores de impresión y otros servidores de propósito especifico. Varias redes de área local, están frecuentemente interconectadas para formar interredes, y estas pueden contener varios miles de computadoras que forman un solo sistema distribuido, habilitando recursos que pueden ser compartidos entre todas las computadoras del sistema.

Tolerancia a fallos. En ocasiones, los sistemas de computación fallan, cuando esto ocurre los programas producirán resultados incorrectos o se pararan antes de que acaben de realizar la operación.

El diseño de sistemas que sean tolerantes de fallos, esta basado en dos características:

Redundancia de hardware: El uso de componentes redundantes;
Recuperación de software: El diseño de programas para recuperase de fallas.

La distribución del hardware redundante requerido para la tolerancia de fallos puede ser diseñada de manera que el hardware sea explotado por actividades no criticas cuando no se presentan fallas. Por ejemplo, una base de datos puede ser duplicada en varios servidores para asegurar que la información permanezca accesible después de que falle un servidor. Los servidores pueden ser diseñados para detectar fallas en otros servidores; cuando una falla es detectada en un servidor, los clientes son dirigidos a los servidores que quedan.

La recuperación del software, envuelve el diseño del mismo, de manera que el estado de la información permanente, pueda ser recuperada cuando una falla sea recuperada. En general, las operaciones ejecutadas por algunos programas, pueden quedar incompletas cuando ocurre una falla, y la información que actualizan puede no quedar en un estado estable.

Los sistemas distribuidos poseen además un alto grado de disponibilidad para encarar fallos de hardware.

Transparencia: La transparencia, se refiere al hecho de que el sistema sea percibido como un todo en lugar de una colección de componentes individuales. La separación de los componentes, es una propiedad heredada de los sistemas distribuidos. La separación permite la ejecución paralela de programas, la recuperación de fallas sin interrumpir el sistema completo.

Entre las ventajas de los sistemas distribuidos podemos mencionar las siguientes:

- Datos compartidos. Permite que los usuarios tengan acceso a la misma información y a aplicaciones comunes.

- Dispositivos compartidos. Permite que varios usuarios compartan periféricos.

- Comunicación. Facilita la comunicación entre los usuarios conectados al sistema. (correo electrónico)

- Es flexible, ya que difunde la carga de trabajo entre las maquinas disponibles en la forma mas eficaz en cuanto a costos.

Las desventajas de los sistemas distribuidos son:

- Software. No existe mucho software para ser usado por los sistemas distribuidos.

- Red. Se corre el riesgo de que se sature la red o de que se pierdan mensajes.

- Seguridad. Debido a que se puede tener acceso a la mayoría de los datos, en ocasiones se puede tener acceso a información que no debe ser consultada por algunos usuarios.

2.3.1 Elementos del modelo de procesamiento distribuido

El procesamiento distribuido utiliza servidores de archivos, estaciones de trabajo, tarjetas de red, nodos, repetidores, puentes, ruteadores, y pasarelas.

El propósito del servidor de archivos es el de distribuir archivos y segmentos de programas a las estaciones de trabajo, imprimir desde una ubicación central y controlar el flujo de conexión entre estaciones de trabajo. Mas del 90% del procesamiento se realiza a nivel de la estación de trabajo, dejando entre el 5 y el 10% de carga, para tareas de administración en el servidor de archivos.

Estaciones de trabajo. Una computadora personal, ademas de utilizarse como servidor de archivos, puede servir también como estación de trabajo Linux. Linux se diseño para ejecutarse en una configuración mínima de hardware.

Generalmente los recursos se deben de aplicar a nivel de estación de trabajo, donde se realiza la mayor parte del procesamiento. La mayor parte de los recursos adicionales dependen del tipo de tareas que se tenga pensado realizar.

Tarjetas de red. Es el enlace físico entre la computadora y el cableado de la red. Estas tarjetas están comúnmente disponibles para cables coaxiales, y para cables de par trenzado.

Nodos. Un nódulo sirve como punto de conexión de los cables y puede ser activo o pasivo. Un nódulo pasivo normalmente tiene cuatro conectores. Un nódulo activo normalmente tiene ocho puertos y amplifica o retransmite la señal.

Repetidores. Estos amplifican o regeneran la señal en la red de modo que se puedan ampliar las limitaciones normales de distancia del cableado de red.

Puentes. Se utiliza un puente cuando se requiere conectar dos tipos de red similares.

Rotures. Los routers se utilizan en redes complejas y de gran tamaño donde hay muchos caminos para que las señales de la red viajen al mismo destino. El router determina y envía la señal por la ruta mas efectiva.

Pasarelas. Las pasarelas se utilizan cuando se quiere conectar tipos de redes no similares. La pasarela realiza las conversiones de protocolo necesarias de modo que las dos redes se puedan comunicar.

2.3.2 REVISION DE LAS TOPOLOGIAS

La palabra topología se refiere al modo en que las estaciones de trabajo y los servidores de archivos se conectan en una red. El nombre de varias de las topologías proviene del patrón que hacen los cables después de que se conectan varias terminales, estaciones de trabajo y servidores de archivos. Las topologías mas comunes son estrella, bus y anillo. Cuando se utiliza mas de una topología en la red, esta queda referida como una red híbrida.

Topología en Bus o Arbol.

El principio de esta red, es la ausencia de una computadora central. Cada nodo esta conectado a un medio único y pasivo de comunicaciones por medio de unidades de interfaz y derivadores. Esta topología permite que los mensajes sean transmitidos a todos los nodos simultáneamente a través del Bus (ver fig. 2.3). Como consecuencia de que cada estación conectada es independiente, ya que no requiere de que uno o varios dispositivos intermedios le transmitan el mensaje, aumentan notablemente la confiabilidad de la red, pero cada nodo debe ser capaz de transmitir, recibir y resolver problemas.

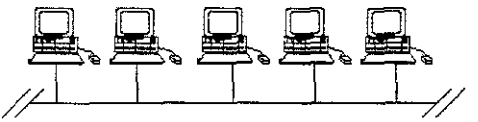

Fig. 2.3 Topología Bus

La topología de bus constituye la base de los buses Ethernet y del Token.

La topología en Bus suele ser relativamente sencilla, se usa normalmente en redes muy pequeñas o que tienen muy poco trafico. La respuesta es excelente cuando hay poco trafico pero a medida que aumenta la carga, la respuesta disminuye rápidamente.

El fallo de una estación no afecta normalmente a la red. Las redes en bus son vulnerables a los falos del canal principal y a otros problemas que afectan al canal de comunicación. Cuando se producen anomalías su causa es muy difícil de localizar; sin embargo, una vez localizados son bastante fáciles de reparar.

La expansión y reconfiguracion de red es muy sencilla, cualquier dispositivo que se desee cambiar o instalar se puede conectar en el punto mas adecuado sin tener que cambiar nada en el resto de la red, aunque resulta difícil utilizar microordenadores y dispositivos de fabricantes diferentes.

En resumen como ventajas y desventajas de esta topología podemos mencionar las siguientes:

- El medio de transmisión es totalmente pasivo.
- Es sencillo conectar nuevos dispositivos.
- Se puede utilizar toda la capacidad de transmisión disponible.
- Es fácil de instalar

Las desventajas son:

- La red en si es fácil de intervenir con el equipo adecuado, sin perturbar el funcionamiento de la misma.
- El interfaz con el medio de transmisión ha de realizarse por medio de dispositivos inteligentes.
- Los dispositivos no inteligentes necesitan unidades de interfaz muy sofisticados.
- En ocasiones, los mensajes interfieren entre si.
- La longitud del medio de transmisión no sobrepasa generalmente los 2 mil metros.

Topología de Anillo

La red en anillo forma un circulo de conexiones punto a punto de estaciones contiguas. Los mensajes van de un dispositivo a otro hasta llegar al adecuado. Las estaciones están conectadas al cable por medio de una unidad de acceso, que a su vez, esta conectado a un repetidor al que se le conoce como Multistation Acces Unit (MAU), el cual transmite las señales que van dirigidas a otros nodos (Ver figura 2.4).

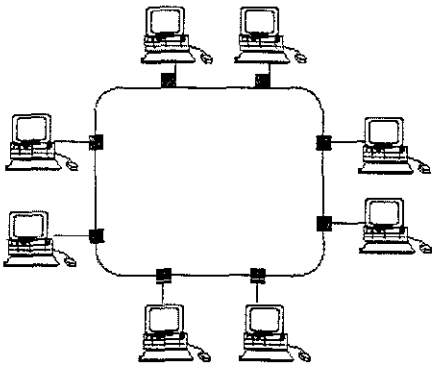

Fig. 2 4 Topologia Anillo.

La red Token Ring de IBM es un ejemplo de esta topología.

Una red en anillo es de gran utilidad en situaciones en que se ha de asignar la capacidad de la red de forma equitativa o cuando haya que conectar un pequeño numero de estaciones que funcionen a velocidades muy altas en distancias muy cortas. Una red en anillo requiere de hardware complicado. El desvío de mensajes es en gran medida sencillo; puesto que la señal solamente se mueve en una dirección, la estación emisora solo necesita saber la dirección de la receptora.

Con trafico muy alto la respuesta del sistema permanece bastante estable. El aumento del tiempo de espera es menor que en otros tipos de red; sin embargo, el tiempo de espera medio es bastante alto incluso cuando la carga del sistema es baja.

El fallo de una sola estación o de un canal puede hacer que se afecte todo el sistema. Esto es debido a la interdependencia de las estaciones. En este tipo de topología resulta bastante difícil localizar un fallo; en un sistema muy amplio puede no ser posible reparar inmediatamente el sistema. Si se desea mantener la red en funcionamiento; es necesario duplicar los recursos o utilizar un método para evitar los puntos en los que se ha producido el percance.

En una red en anillo equipada con centros conectores apropiados es bastante sencillo añadir o suprimir estaciones sin tener que hace un gran numero de conexiones, lo que abarata los costos.

En resumen, podemos decir que las ventajas y desventajas son las siguientes:

Ventajas:

- La capacidad de transmisión se reparte equitativamente entre todos los usuarios.
- La red no depende de un nodo central.
- Se simplifica al máximo la distribución de mensajes.
- Resulta fácil enviar un mismo mensaje a todas las estaciones.
- El tiempo de acceso es moderado
- El índice de error es muy pequeño
- Se pueden conseguir velocidades de transmisión muy altas.
- Permite utilizar distintos medios de transmisión.

Desventajas:

- La fiabilidad depende de los repetidores
- Es necesario un dispositivo monitor
- La instalación es bastante complicada.

Topología Estrella

En una configuración en estrella cada estación de trabajo esta conectada a un nodo central por medio de un canal punto a punto dedicado. Las estaciones pasan los mensajes al servidor central, y esto lo retransmite a la que vaya dirigido. Con esta topología se pueden tener Nodos activos o pasivos (Ver figura 2.5).

Un nódulo pasivo es simplemente un punto de conexión para las estaciones de trabajo. Un eje activo también ofrece amplificación de la señal. Starlan de AT&T es un ejemplo de red que utiliza la topología en estrella.

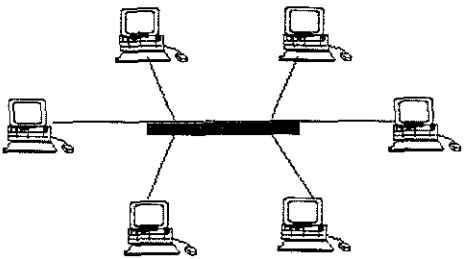

Fig 2.5 Topología Estrella

El control de la red se puede asignar de las tres maneras siguientes:

1. El control reside en el nodo central el cual efectúa la retransmisión de mensajes. Los datos recibidos en la estación central pueden ser procesados o dentro de la misma o pueden ser enviados a otra para que los procesen.

2. El control puede estar a cargo de una de las estaciones exteriores, en vez del nodo central. El gestor actúa de conmutador estableciendo conexiones entre las distintas estaciones.

3. El control puede estar distribuido entre todas las estaciones. El nodo se usa para enviar mensajes a sus destinos y para resolver sus solicitudes de conexión.

En los tres casos el nodo central es la estación principal, si falla, se para toda la red. La estación central proporciona un punto lógico para conectar directamente los recursos compartidos más importantes.

El tamaño y la capacidad de la red están directamente relacionados con la potencia de la estación central. La carga que conlleva todo lo relacionado con la compatibilidad la soporta el nodo central.

Actualmente la red en estrella es la mejor forma de integrar servicios de datos y voz. Una red de datos con esta topología, que utilice los nuevos sistemas PBX digitales ofrece las ventajas y ahorro de los servicios telefónicos. La configuración en estrella puede ser bastante complicada; las estaciones conectadas a la central, pueden, a su vez actuar de nodo central para otras, o pueden estar conectadas a enlaces de comunicaciones remotas.

Esta topología es buena para una carga moderada del sistema. Sin embargo, el tamaño y capacidad de la red y, por tanto, la respuesta están directamente relacionadas con la potencia del nodo central. La dependencia de la red es muy alta; normalmente la estación no se puede usar para ninguna otra cosa mientras se esta actuando como controlador de red.

La fiabilidad de la red depende completamente del nodo central. Si este falla cesa toda la actividad de la red. El fallo de una sola estación de trabajo no afecta el funcionamiento del sistema. En cualquier caso, la identificación y reparación de problemas quedan simplificadas por el control centralizado.

La expansión de esta topología es muy restringida, ya que la mayoría de los nodos centrales solo pueden soportar un numero limitado de interfaces de red. A menudo, al usuario se le imponen limitaciones de ancho de banda y de velocidades de transmisión.

Como ventajas y desventajas de este tipo de topología están:

Ventajas:

- Es ideal en configuraciones en las que hay que conectar muchas estaciones a una.
- Se pueden conectar terminales no inteligentes
- Las estaciones pueden tener velocidades de transmisión diferentes.
- Permite utilizar distintos medios de transmisión.
- Se puede obtener un alto nivel de seguridad.
- Es fácil de detectar los errores.
- La transmisión de los mensajes esta controlada por el nodo central.

Desventajas:

- Es susceptible de averías en el nodo central.
- Elevado precio debido a la complejidad de la tecnología que se necesita en el nodo central.
- La instalación de los cables resulta bastante cara.
- La actividad que debe soportar el nodo central hace que normalmente las velocidades de transmisión sean inferiores a las que se consiguen en las topologías de Bus y en Anillo.

2.4 MODELO CLIENTE SERVIDOR

Pare entender los conceptos de cliente/servidor se debe de entender que describen una relación lógica entre una parte que solicita un servicio (el cliente) de otra parte (el servidor) que provee el

servicio compartido solicitado. Las partes del cliente y del servidor pueden existir o no físicamente en diferentes computadoras. Un cliente puede tener una relación con diferentes servidores y un servidor puede atender las solicitudes de múltiples clientes. La relación entre un cliente y un servidor es llevada a cabo por medios de transacciones consistentes de solicitudes y respuestas bien definidas. Lo sobresaliente del modelo cliente/servidor es que el cliente y el servidor comparten la carga de trabajo entre ellos.

El termino pareja, en un ambiente de comunicación pareja a pareja, se refiere a dos entidades con buena relación. Cada pareja entiende el protocolo usado por sus parejas y participa en la comunicación como se muestra en la figura 2.6.

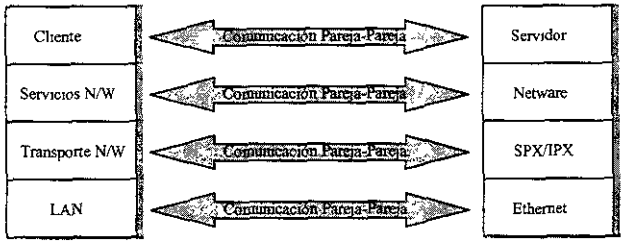

Fig. 2.6 Relación de comunicación entre el cliente y el servidor.

Procesos cliente /servidor

Los procesos del cliente mandan peticiones a los procesos del servidor, El cual responde a dichas peticiones. Como su nombre lo indica, el servidor de procesos, provee los servicios a sus clientes, generalmente por medio de procesos específicos, que solo este puede ejecutar. El proceso del cliente, liberado de la complejidad y sobrecarga de procesar la petición, es capaz de realizar otras tareas productivas. La interacción entre los procesos del cliente y del servidor es un intercambio, en el cual el cliente es proactivo y el servidor es reactivo.

En un ambiente real cliente/servidor, tanto los procesos del cliente como los del servidor son ndiferentes ya sea si se ejecutan en una misma computadora o en diferentes. Algunos protocolos de

red tal como el LAN SERVER de IBM no soportan que los procesos del cliente y del servidor estén en el mismo sistema. Sin embargo, esta es tan solo la restricción de un protocolo en particular, y no una característica general del modelo cliente/servidor. Desde una perspectiva teórica, el tamaño de la computadora no importa. El proceso del servidor puede ser ejecutado en una computadora de menor capacidad que la que ejecuta el proceso del cliente. La figura 2.7 muestra varios tipos de sistemas cliente/servidor.

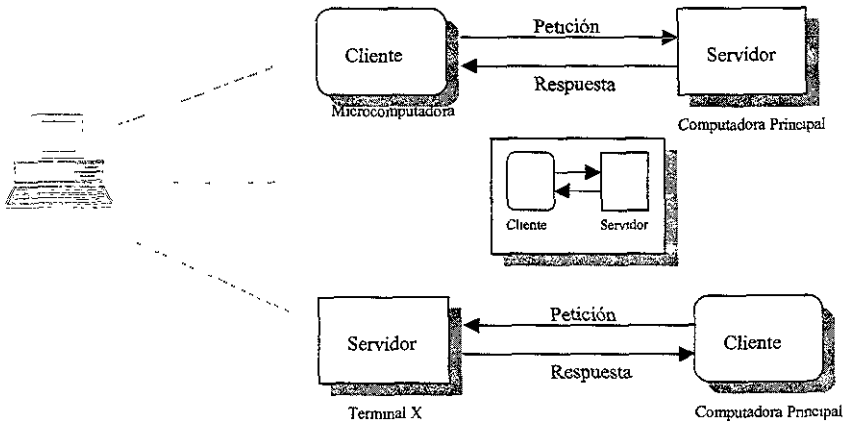

Fig 2.7 Tipos de sistemas cliente/servidor.

Los sistemas cliente/servidor son mas interesantes cuando los procesos del cliente y del servidor se ejecutan en computadoras diferentes conectadas mediante una red, ya sea una LAN o una WAN, aunque las redes locales son los tipos mas comunes del modelo cliente/servidor.

Existe la confusión de que la tecnología cliente/servidor son las bases de datos SQL(Structured Qyery Language). En realidad, la mayoría de los procesos del cliente usan un proceso SQL para acceder a la base de datos a través de la red, pero este es tan solo un tipo del modelo cliente/servidor. Es posible desarrollar una gran variedad de aplicaciones cliente/servidor que no utilicen SQL.

Atributos del cliente

El proceso del cliente es proactivo, cuando genera peticiones al servidor. Un cliente puede interactuar con un solo servidor o con varios servidores para lograr su trabajo. Sin embargo, al menos un proceso del servidor es necesario.

En el nivel de aplicación, el cliente es responsable del mantenimiento y proceso de la comunicación son el usuario. Entre la tareas se encuentran las siguientes:

- Manejo de la pantalla
- Interpretación de comandos
- Entrada y validación de datos
- Recuperación de errores.

En aplicaciones gráficas se incluye:

- Manejo de la ventana
- Control de ratón y el teclado
- Control de las cajas de dialogo
- Administración del sonido y vídeo.

El proceso del servidor es reactivo, accionado por la petición de sus clientes. El servidor se esta ejecutando siempre, proviendo de servicios a varios clientes. Estos servicios pueden ser proporcionados, ya sea directamente por el servidor, o indirectamente como esclavo, cuando el proceso lo generan otros servidores. La figura 2.8 muestra el uso de servidores esclavo y maestro.

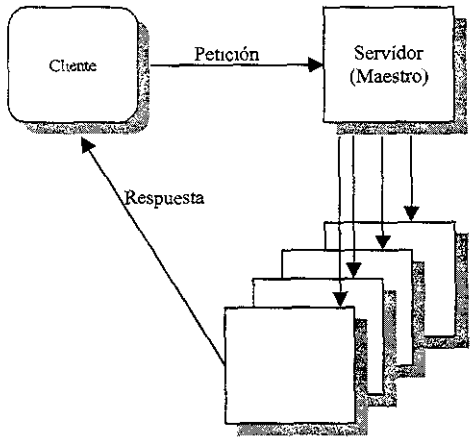

Fig 2.8 Uso de servidores Esclavo y Maestro

Un servidor es de función especifica; ejecuta un conjunto predefinido de transacciones. Existen múltiples servidores para proveer de un juego de funciones especificas a los clientes. Potencialmente, múltiples clientes pueden accesar a un servidor al mismo tiempo.

Un servidor ejecuta toda la lógica requerida para procesar una petición y no interactua con otros servidores usualmente, la petición es tratada como una transacción (que es una unidad indivisible de trabajo). Por ejemplo, si la petición es de agregar un nuevo envío a la base de datos, el servidor tendrá que actualizar varias tablas en la base de datos para mantener la integridad de la información de la base de datos.

Si el cliente usa varios servidores, es responsabilidad del cliente llamar a los servidores cuando sean requeridos. Los procesos del servidor, incluyen todos los que se relacionan con acceso, almacenaje y actualización de la información compartida; actualización de la información almacenada; y cualquier otra administración de recursos compartidos. Los recursos compartidos pueden ser información, procesador, disco de almacenaje, impresoras y comunicaciones.

El resultado del desarrollo del procesamiento de datos distribuido es un modelo cliente/servidor. Linux puede ser usado como cliente y servidor.

Para ejemplificar una configuración cliente/servidor, podemos asumir que varias estaciones de trabajo Linux (Los clientes) están conectados a un servidor (que ejecuta Linux) por medio de una topología de bus. El servidor tiene directorios para cada cliente, donde pueden almacenarse los archivos importantes, de los que se hará copia con la copia de seguridad de cada noche. El servidor tiene también directorios donde los clientes pueden compartir archivos. Están conectados al sistema una impresora láser.

CAPITULO 3

ADMINISTRACION DE UNA RED

3 ADMINISTRACION DE UNA RED

Un sistema Linux precisa de una configuración inicial y luego de una atención continua para asegurar que el sistema se mantenga efectivo, fiable, y eficiente para todos los usuarios. El administrador del sistema es la persona responsable de atender las necesidades del sistema

Un sistema Linux debe tener una o más personas designadas como administradores del sistema para gestionar el sistema y prever su rendimiento. El administrador del sistema tiene la responsabilidad del funcionamiento adecuado del sistema.

Todos los sistemas UNIX son distintos de una forma u otra y cada sistema individual UNIX es distinto en la forma de administrarse. Linux no es una excepción. Las tareas de administración varían, dependiendo, entre otras cosas, del numero de usuarios a administrar, los tipos de periféricos conectados a la computadora (impresoras, unidades de cinta, etc.), las conexiones de red y el nivel de seguridad necesaria.

Un administrador del sistema, tiene el poder y la responsabilidad de establecer y mantener un sistema que proporcioné servicio fiable y efectivo. En un entorno multiusuario hay un cierto numero de objetos y prioridades.

Todos los sistemas Linux tienen un solo usuario que puede realizar prácticamente cualquier operación. A este usuario se le denomina superusuario y tiene un nombre especial de entrada llamado root. Cuando el usuario root entra en el sistema lo hace en el directorio raíz del sistema de archivos.

El administrador del sistema entra como superusuario para realizar tareas privilegiadas. Para el trabajo normal en el sistema, el administrador entra como un usuario normal. El nombre de entrada del superusuario (root) tiene propósitos limitados y especiales. El numero de usuarios que pueden entrar como root deben de ser los menos posibles. Cuando

una persona entra como root, se convierte en superusuario y tiene privilegios absolutos en el sistema. Con este privilegio puede cambiar los atributos de cualquier archivo, iniciar el sistema, hacer copias de seguridad de los datos del sistema y muchas tareas más.

Entre las tareas que debe de realizar un administrador de sistema se consideran las siguientes:

- Administrar usuarios. Dar de alta a usuarios, darlos de baja y modificar sus posibilidades y privilegios.

- Configurar dispositivos. Hacer disponibles y compartir dispositivos como, por ejemplo, impresoras, terminales, modems, unidades de cinta, etc.

- Hacer copias de seguridad. Programar, hacer y almacenar copias de seguridad para poderlas restaurar si se dañan o pierden los archivos del sistema.

- Formar usuarios. Porporcionar directa o indirectamente formación a los usuarios de modo que puedan utilizar el sistema de forma efectiva y eficiente.

- Asegurar el sistema. Evitar que los usuarios interfieran unos con otros a través de acciones accidentadas o deliberadas.

- Registrar los cambios del sistema. Mantener un libro para registrar cualquier actividad significativa que se refiera al sistema.

3.1 CONJUNTO DE PROTOCOLOS TCP/IP

El conjunto de protocolos ampliamente utilizados conocidos como Transmision Control Protocol (TCP/IP) es cada vez más importante ya que de él dependen importantes redes como Internet.

TCP/IP surgió como un proyecto promovido por el gobierno hasta alcanzar su extenso uso actual, conectando redes de todos los tamaños. Reconocido por la capacidad para permitir comunicaciones entre diferentes equipos, se encuentra virtualmente en todas las estaciones de trabajo y computadoras.

Características principales de IP

Las funciones de IP son:

- Encapsulado de datos y formato del encabezado.
- Rutado de datos a través de la red.
- Transferir datos a otros protocolos.
- Fragmentación y ensamble.

3.1.1 HISTORIA DE TCP/IP

El protocolo TCP/IP fue desarrollado por departamento de defensa del gobierno de los Estados Unidos. El proyecto fue iniciado en respuesta a la necesidad de tener todas las computadoras del departamento de defensa y del gobierno en una sola red. Esta configuración hizo posible que toda la información necesaria estuviera al alcance del personal autorizado; sin importar en que parte de la red se encontrara.

Debido a la magnitud de la diversidad de plataformas de software o hardware que el gobierno y las universidades tenían, el departamento de defensa considero muy costoso tener todas estas plataformas diferentes conectadas a una red global. Como resultado, en 1969 el 'Defense Advanced Research Project Agency' fue asignado para desarrollar una red

que pudiera ser utilizada con varias plataformas. La red ARPANET fue construida para uso en el desarrollo y pruebas de tecnologías.

A mediados de los años 70, el Departamento de Defensa (DOD) de los Estados Unidos reconoció el desarrollo de un problema de comunicaciones electrónicas dentro de su organización. La comunicación de la información electrónica entre el personal del DOD, laboratorios de investigación y universidades se enfrentaban a un importante obstáculo. Las diferentes entidades tenían hardware procedente de diferentes fabricantes, que ejecutaban sistemas operativos diferentes y utilizaban diferentes topologías y protocolos de red.

Se le pidió a la agencia de investigación de proyectos avanzados (ARPA) que resolviera el problema que suponía el tratar con diferentes equipos y topologías de red. ARPA formo una alianza con universidades y fabricantes de sistemas para el desarrollo de estándares de comunicación. Esta alianza desarrollo una red de cuatro nodos, que es la base de la red Internet actual. Durante los años 70 esta red emigro a un nuevo diseño de protocolo central que se convirtió en la base de TCP/IP.

A inicios de los 80's todas las redes de desarrollo interconectadas fueron convertidas al protocolo TCP/IP, y ARPANET se convirtió en el backbone de Internet.

3.1.2 MODELO DE INTERCONEXION DE SISTEMAS ABIERTOS (OSI)

En la actualidad se utilizan muchos tipos diferentes de computadoras. Estos sistemas se diferencian entre sí por sus sistemas operativos, CPU, interfaces de red y muchas otras variables. Estas diferencias hacen que el problema de comunicación entre diferentes sistemas sea importante. En 1977, la organización internacional para la normalización de estándares (ISO), creo un subcomite con el fin de desarrollar estándares de comunicaciones de datos para promover la interoperabilidad multifabricante. El resultado de esto es el modelo de Interconexión de Sistemas Abiertos mejor conocido como OSI (Open Systems Interconection).

OSI es un modelo de arquitectura de red por capas que incorpora las ideas y experiencias de muchos de los primeros diseños, fue propuesto por la Organización de Normas Internacionales (ISO, International Standards Organization). El modelo OSI no especifica ningún estándar o protocolo de comunicaciones sino que, en su lugar, proporciona normativas a seguir por las tareas de comunicaciones. El modelo OSI tiene siete capas, representadas en la figura 3.1. A continuación daremos una breve descripción de cada capa.

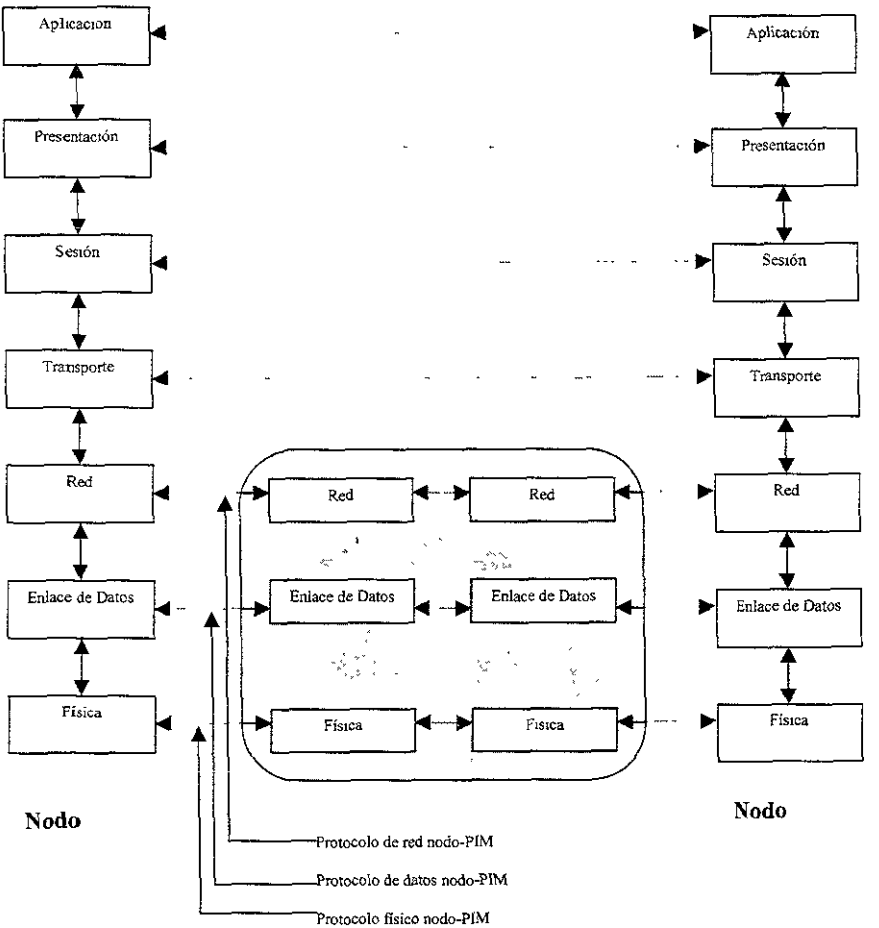

Fig. 3.1 Modelo OSI

A cada capa se le asigna un conjunto determinado de funciones. Cada capa utiliza los servicios de la capa que se encuentra debajo de la suya y proporciona servicios a la capa que se encuentra encima. Por ejemplo, la capa Red utiliza los servicios de la capa de Enlace de datos y proporciona servicios relacionados con la red a la capa de Transporte a continuación se describen los servicios ofrecidos por cada capa:

Física (Capa 1) Esta capa proporciona la conexión física entre un sistema informático y la red. Especifica las asignaciones de conector y pin, niveles de voltaje, etc.

La capa física esta encargada de transmitir flujos de bits a través del canal de comunicación. Se ocupa principalmente de los circuitos de comunicación y de sus interfaces físicas y procedimientales con el medio de transmisión físico subyacente.

Enlace de datos (Capa 2) Esta capa empaqueta y desempaqueta los datos para su transmisión. Forma la información en tramas. Una trama representa la estructura exacta de los datos físicamente transmitidos por cable u otro medio.

La capa de enlace de datos toma la facilidad de transmisión de flujo de bits en bruto y la mejora para proporcionar aparentes líneas de comunicación libres de errores entre computadoras que están directamente conectados. Debido a las diferencias en las implementaciones de subredes de comunicación descritas anteriormente, en las redes de tipo "almacenar y reexpedir" la capa de enlace de datos opera sobre conexiones PMI-PMI y nodo-PMI, y en las LAN gestiona las comunicaciones nodo-nodo.

La abstracción de una línea libre de errores se consigue dividiendo los flujos de bits en franjas (frames) y añadiendo bits adicionales para detección de errores. La capa de enlace de datos procesa también las franjas de reconocimiento enviadas de vuelta al receptor. La

perdida de reconocimientos o su mutilación por transmisión defectuosa pueden dar lugar a la generación de franjas duplicadas. La capa de enlace de datos esta encargada de resolver los problemas relativos a mensajes dañados, perdidos y duplicados. También esta encargada del control de flujo, un mecanismo diseñado para evitar el desbordamiento de los nodos que puede resultar de, entre otras razones, las discrepancias de velocidad entre emisores y receptores.

Red (Capa 3) Esta capa proporciona el direccionamiento de los datos por la red.

La capa de red controla la operación de la subred de comunicación. Sus funciones principales son el encadenamiento de paquetes, el mantenimiento y el control de congestión. Las consideraciones y problemas entre redes resultantes de la reunión de redes heterogéneas están también confiadas a la capa de red. Estas pueden incluir conversiones entre diferentes esquemas de direccionamiento y diferentes tamaños de paquetes.

En muchas redes de difusión y en las LAN, la capa de red es delgada o puede faltar, ya que tiene poco que hacer.

Transporte (Capa 4) Esta capa proporciona la secuencia y acuse de recibo de la transmisión.

La tarea de la capa de transporte es proporcionar transporte de mensajes independiente de la red entre pares de extremos de la red, o puertos. Como indica la figura 3.1, la capa de transporte es la primera que proporciona una conexión verdadera entre fuente y destino. En las capas inferiores, la comunicación se efectúa entre una maquina y sus vecinos inmediatos, y no necesariamente entre los nodos fuente y destino comprometidos en una conversación.

La capa de transporte acepta datos procedentes de la capa de sesión, los divide en unidades, más pequeñas tales como paquetes si es necesario y asegura que todas las piezas sean reunidas adecuadamente en el extremo receptor. El transporte efectivo de los trozos los

efectúa la capa de red.

La capa de transporte soporta dos modos de comunicación:

- Circuito virtual
- Datagrama

Un circuito virtual se asemeja a un sistema telefónico. Es un canal de comunicación lógica establecida entre dos nodos con objeto de mantener una conversación. Un circuito virtual entrega fiablemente mensajes en el orden en que son enviados. El modo de comunicación que utiliza circuitos virtuales se denomina servicio orientado a conexión ya que requiere que las entidades interesadas establezcan implícitamente un enlace de comunicación con objeto de intercambiar los mensajes.

Un datagrama se considera un mecanismo no fiable de entrega de mensajes, ya que no reconoce la recepción de los mensajes. El servicio de datagrama se asemeja al servicio postal, en donde los mensajes individuales son enviados independientemente y sin acuerdo previo. Debido a esta analogía, los mensajes enviados como datagramas pueden ser perdidos o recibidos fuera de orden. El servicio de datagramas tiene menos recargo que el circuito virtual y generalmente es más rápido ya que no requiere la preparación del enlace. Los datagramas son populares en redes de área local cuya fiabilidad suele ser lo suficientemente buena para reducir las preocupaciones con respecto a perdidas de mensajes.

Sesión (Capa 5) Esta capa establece y finaliza los enlaces de comunicación.

La capa de sesión permite que los procesos residentes en nodos diferentes se comuniquen entre sí, esta capa establece sesiones entre procesos que proporcionan transporte ordinario de datos y algunos servicios adicionales tales como aperturas de sesiones remotas y transferencia de archivos.

La capa de sesión también esta encargada de proporcionar sincronización y gestión de testigos para soportar interacciones entre procesos a través de los circuitos virtuales que establece.

Presentación (Capa 6) Esta capa efectúa la conversión de los datos y se asegura de que los datos se intercambian a un formato universal.

En vez de tan solo trasladar datos entre las entidades de los extremos, la capa de presentación efectúa algunas funciones habituales que pueden requerir conocimiento de sintaxis y la semántica de la información transmitida. Un ejemplo típico es la codificación de los datos en algún formato estándar, independientemente de la maquina. Esta codificación permite conversiones de formatos de datos, tales como ordenación de bytes y representación de punto flotante, para permitir la comunicación entre maquinas heterogéneas. Además, la capa de presentación proporciona, opcionalmente, cifrado y compresión de datos. La decodificación necesaria es efectuada por la capa de presentación del extremo receptor.

Aplicación (Capa 7) Esta capa proporciona una interfaz a la aplicación que ejecuta un usuario: una pasarela entre las aplicaciones de usuario y el proceso de comunicación de red.

La capa de aplicación proporciona una variedad de protocolos habitualmente requeridos por los procesos de aplicación que corren en computadoras separadas a cuenta de las tareas de usuario. Los protocolos pueden incluir correo electrónico, admisión de trabajos remotos. y transferencia de archivos que ocultan las posibles diferencias de denominación y representación entre los usuarios de los extremos. La capa de aplicación también proporciona una abstracción de terminal denominada terminal virtual de red. Esto permite a los proveedores de aplicaciones escribir código por ejemplo un editor de pantalla, para un tipo de terminal única (la terminal virtual) y confiar en la capa de aplicación para traducir las ordenes relevantes por secuencias de control apropiadas para el tipo o tipos de

terminales locales específicas.

Cada capa se comunica con su igual en otras computadoras. Por ejemplo, la capa 3 en un sistema, se comunica con la capa 3 en otra computadora.

Cuando se pasa la información desde una capa a la siguiente capa inferior, se añade una cabecera a los datos a fin de indicar de donde proviene la información y a donde va. La cabecera mas el bloque de información de datos de una capa se convierten en datos para la siguiente. Por ejemplo, la capa 4 añade su propia cabecera cuando traspasa información a la capa 3. Cuando la capa 3 traspasa la información a la capa 2, la capa remitente interpreta la cabecera mas datos procedentes de la capa 4, como datos y añade su propia cabecera, antes de traspasar esa combinación hacia abajo.

La tabla 3.1 enlista los nombres que recibe la información en cada capa.

Capa OSI	Nombre de la Unidad de información
Aplicación	Mensaje-
Transporte	Segmento
Red	Datagrama
Enlace de datos	Trama o Paquete
Física	Bit

Tabla 3.1 Nombres de la información en cada capa

Con anterioridad a la adopción del modelo OSI, el departamento de Defensa de los Estados Unidos definió su propio modelo de red, conocido como el modelo DOD (Casi siempre llamado Internet). El modelo DOD esta estrechamente relacionado al conjunto de protocolos TCP/IP.

La pila de protocolos TCP/IP representa una arquitectura de red similar al modelo de red OSI de ISO. La figura 3.2 muestra el mapa de las capas TCP/IP en la pila de protocolos ISO.

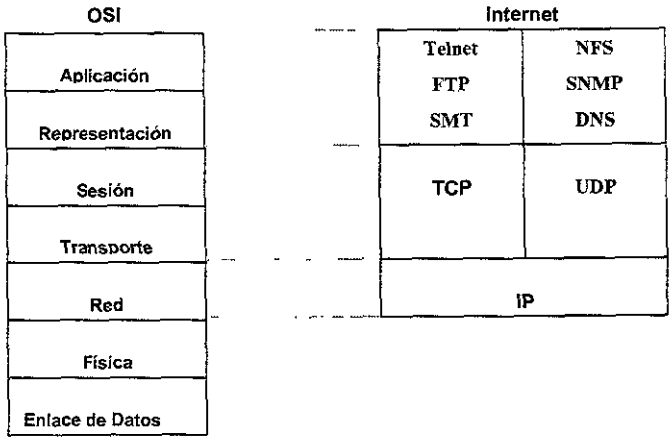

Fig. 3.2 Comparacion de OSI y TCP/IP

TCP/IP no establece tantas distinciones como OSI entre las capas superiores de la pila de protocolos. Las tres capas superiores OSI vienen a ser el equivalente de los protocolos de proceso de Internet. Algunos ejemplos de protocolos de proceso son Telnet, FTP, SMNP, y DNS.

La capa de Transporte del modelo OSI es responsable de la entrega fiable de datos. En la pila de protocolos de Internet, esto corresponde a los protocolos sistema principal a sistema principal. Los ejemplos de estos son TCP y UDP. TCP se utiliza para traducir mensajes de longitud variable procedentes de los protocolos de capa superior y proporciona el necesario acuse de recibo y control de flujo orientado a conexiones entre sistemas remotos.

UDP es similar a TCP salvo que no esta orientado a conexiones y no acusa recibo de los datos. UDP solo recibe mensajes y los transmite a los protocolos de nivel superior. UDP proporciona una interfaz mucho más eficaz para acciones como servicios remotos de discos, debido a que no admite las cargas relacionadas con TCP.

El protocolo Internet (IP) es responsable de comunicaciones sin conexiones entre sistemas.

Se asigna en el modelo OSI como parte de la capa de Red. El modelo OSI es responsable del movimiento de información en la red. Esto se consigue examinando la dirección de la capa de red. Esta dirección determina los sistemas y la ruta a utilizar para enviar el mensaje.

IP proporciona la misma funcionalidad que la capa de Red y ayuda a enviar mensajes entre sistemas. IP puede también fragmentar los mensajes en pedazos y volver a unirlos en su destino. Cada uno de los fragmentos puede tomar una ruta diferente de red entre sistemas. Si el fragmento llega fuera de orden, IP reconstruye los paquetes en su secuencia correcta en destino.

Encapsulado de datos TCP/IP

El protocolo TCP/IP se comunica con su contraparte a través de Internet agregando un encabezado de información a los datos antes de que sean enviados a la capa siguiente. La figura 3.3 ilustra lo que pasa cuando los datos son transmitidos de la capa de aplicación en el nodo A a los protocolos subyacentes para ser entregados en el nodo B.

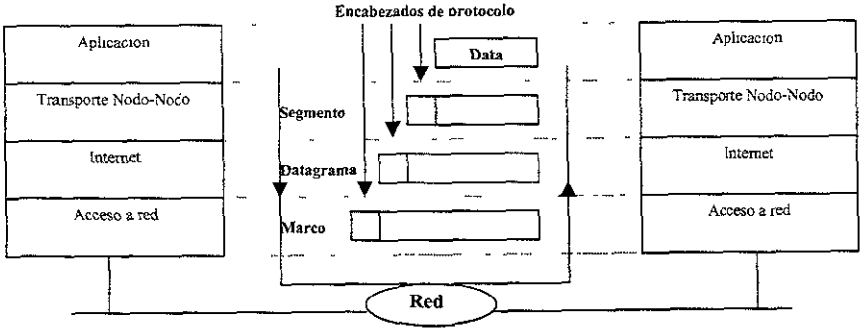

Fig. 3.3 Representacion de los datos cuando se transmiten de la capa de aplicación a capas subyacentes.

Cada capa agrega un encabezado de información que solo tiene relación con su pareja.

A continuación se muestran ejemplos del contenido de cada encabezado.

En la capa de transporte, el encabezado incluye los números de puerto de tanto del destino como del origen. Estos son tratados como números de identificación de proceso, lo que ayuda en el intercambio de datos encapsulados entre procesos designados, sin confundir estos procesos con otros que podrían estar ejecutándose simultáneamente en el mismo nodo. Los datos y el encabezado en esta capa forman una unidad de datos llamada segmento de datos.

En la capa de Internet el encabezado contiene además la dirección IP de los sistemas de comunicación. Los datos y el encabezado en esta capa forman una unidad de datos conocida como Datagrama IP.

En la capa de acceso a red, el encabezado incluye la dirección de control de acceso del destino y del origen que se comunican en la misma red. La unidad de datos en esta capa es llamada marco de datos.

3.1.3 DIRECCIONES IP

En TCP/IP, cada elemento de la red tiene su dirección única, esta dirección es conocida como dirección IP. La dirección IP esta compuesta por 2 partes.

1. La dirección de red, que es común para todos los nodos y dispositivos en la misma red física.

2. La dirección del nodo que es única para el nodo en esa red. El protocolo IP usa esta dirección para rutear la información.

La dirección IP esta organizada como una serie de cuatro octetos. Cada uno de estos octetos define una dirección única, en la que cada parte de la dirección representa una red (y opcionalmente una subred) y la otra parte representa un nodo especifico de la red.

Algunas direcciones tienen significados especiales en Internet, como se describe a continuación.

- Una dirección que empiece con cero hace referencia al nodo local dentro de su red actual. Por ejemplo, 0.0.0.23 hace referencia a la estación de trabajo 23 en la red actual. La dirección 0.0.0.0 hace referencia a la estación de trabajo actual.

- La dirección del bucle interno 127 es importante en procesos de resolución de problemas y diagnosticos de red. La dirección de red 127.0.0.0 es el bucle interno local dentro de una estación de trabajo.

- La dirección ALL es representada por la activación de todos los bits, proporcionando un valor de 255. Por lo tanto, 192.18.255.255 envía un mensaje a todos los nodos de la red 192.18; de la misma forma 255.255.255.255 envía un mensaje a cada nodo de Internet. Es importante utilizar estas direcciones para mensajes de transmisión múltiple y avisos de servicio.

Es importante no utilizar 0, 127 o 255 al asignar números de nodo a sus estaciones de trabajo, ya que estos nodos están reservados y tienen significados especiales

3.1.3.1 CLASES DE DIRECCIONES IP

Las direcciones IP se asignan en rangos llamados clases, dependiendo de la aplicación y del tamaño de la organización. Las clases más comunes son A, B y C. Estas tres clases representan el numero de bits localmente asignable disponibles para la red local. La figura 3.4 muestra los distintos formatos de direcciones correspondientes a cada tipo de las direcciones principales, cada clase de dirección IP se distingue por los primeros bits de la

porción de la red.

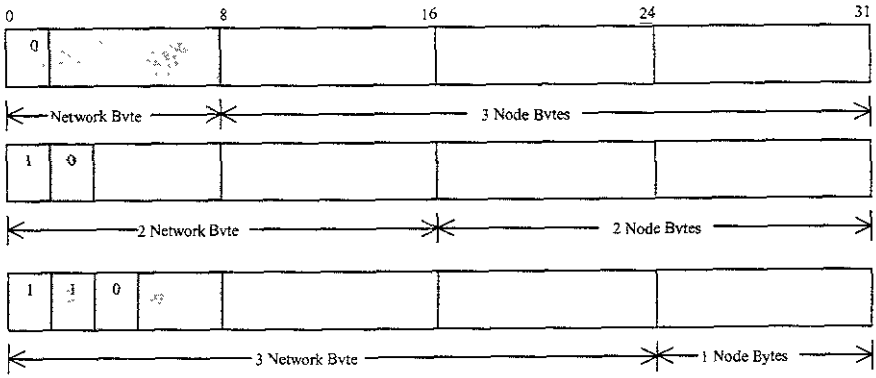

Fig 3.4 Formato de las principales direcciones IP

La tabla 3.2 muestra las relaciones entre las diferentes clases de dirección, el numero disponible de nodos y las configuraciones iniciales de las direcciones.

CLASE	NODOS DISPONIBLES	BITS INICIALES	DIRECCION DE COMIENZO
A	2^{24}=1677772	0xxx	0-127
B	2^{16}=65536	10xx	128-191
C	2^{8}=256	110x	192-223
D		1110	224-239
E		1111	240-255

Tabla 3.2 Relacion entre las diferentes clases de direcciones, el numero de nodos disponibles y las configuraciones iniciales.

Las direcciones de la clase A se utilizan para redes de gran tamaño o para colecciones de redes asociadas. Todas las instituciones educativas están agrupadas bajo una dirección de clase A.

El primer bit se encuentra fijo en 0 y el primer byte, llamado identificador de red, identifica

la red. Los 3 bytes restantes son usados para identificar el nodo en la red. Se puede calcular que hay un máximo de 127 redes de clase A, cada una capaz de alojar millones de nodos. Las direcciones de la clase B se utilizan para redes de gran tamaño con mas de 256 nodos pero con menos de 65536 nodos. Los primeros dos bits están fijos en 10, el primer y segundo byte son usados para identificar la red, y los 2 últimos bytes son usados para identificar el nodo. Puede haber centenares de redes clase B, capaces de alojar centenares de nodos.

Las direcciones de la clase C son utilizadas por casi todas las organizaciones. Es aconsejable que una organización obtenga varias direcciones de la clase C debido que el numero de direcciones de la clase B esta limitado. Los primeros 3 bits se encuentran fijos en 110, los primeros 3 bytes son usados para identificar la red; y el ultimo byte es usado para identificar el nodo. Las redes de clase C son las más pequeñas de todas las clases. Cada red puede alojar 254 nodos (no 256 porque 0x0 y 0xFF están reservadas para otros propósitos) con 3 bytes reservados para identificar la red. Se pueden definir millones de redes clase C.

La clase D esta reservada para los mensajes de transmisión múltiple en la red, mientras que la clase E esta reservada para experimentación y desarrollo. Los primeros 4 bits, están fijos a 1110. Una dirección clase D es una dirección (MULTICAST), que identifican a un grupo de computadoras que pueden estar ejecutando una aplicación distribuida en la red.

Para hacer más sencilla la administración de las direcciones, los administradores de TCP/IP pueden configurar los nodos y los ruteadores, con direcciones que usan notación de punto decimal.

La notación de punto decimal trata las direcciones de 32 bits como 4 bytes separados. Cada byte es representado por su decimal equivalente a un rango de 0 a 255 (que es el rango decimal equivalente para un patrón binario de 8 bits).

3.1.3.2 NOMBRADO DE LA RED

La asignación de nombres a los nodos de la red requiere de planificación. Al seleccionar los nombres, se debe tomar en cuenta la gestión de la red y la aceptación de los usuarios. Muchas organizaciones tienen estándares de nombramiento de red.
La asignación de nombres debe hacerse de forma que proporcione nombres exclusivos a las computadoras.
Al asignar los nombres a la red se deben de tomar en cuenta los siguientes puntos:

- Mantener los nombres sencillos y cortos con un máximo de seis a ocho caracteres. Aunque el protocolo Internet permite la utilización de nombres excesivamente largos. (Cada etiqueta puede tener una longitud de hasta 63 caracteres. Cada parte de un nombre de dominio completo, separado por un punto, es una etiqueta.

- No comenzar el nombre con números.

- No utilizar caracteres especiales en el nombre.

- No duplicar nombres

- Ser coherente en la política de asignación de nombres.

Los nombres de Internet son representativos de las organizaciones y de la funcionalidad de los sistemas dentro de la red. Los nombres en Internet le permiten hacer referencia a un usuario en un nodo determinado como por ejemplo:

Ejemplo@PC1.Programming.compañia.com

Arbol de nombrado NIC

El NIC (Centro de Información de la Red) mantiene un árbol de nombrado para la red. Este árbol se utiliza para agrupar organizaciones bajo ramas similares del árbol Las organizaciones importantes están agrupadas bajo ramas similares. Esta es la fuente de etiquetas Internet como, por ejemplo, com, edu y gob que se pueden observar en algunos nombres Internet.

3.1.4 SUBREDES

El establecimiento de subredes es el proceso de dividir una gran red lógica en redes físicas más pequeñas. Las razones para dividir una red pueden incluir limitaciones eléctricas de la tecnología de trabajo en red, un deseo de segmentar por razones de sencillez ubicando una red distinta en cada planta de un edificio o una necesidad de contar con ubicaciones remotas conectadas por una línea de alta velocidad.

Las redes resultantes son partes más pequeñas de la red completa y son más sencillas de gestionar. Las subredes más pequeñas se comunican entre sí a través de pasarelas y routers. Además, dentro de una organización puede haber varias subredes que se encuentren físicamente en la misma red. Esto se podría hacer con el objetivo de dividir lógicamente las funciones de red en grupos de trabajo.

"Las subredes individuales son una división de la red completa. Supongamos que una red clase B se divide en 64 subredes distintas. Para hacerlo, la dirección IP se visualiza en dos partes: red y sistema principal (Ver figura 3.5). La parte de la red se convierte en la dirección IP asignada y en los bits de información de la subred. Estos bits son en esencia eliminados de la parte del sistema principal. El numero asignado de bits para una red de la clase es 16. La parte e la subred añade 6 bits, teniendo así un total de 22 bits para distinguir la subred. Esta división resulta en 64 redes con 1024 nodos cada uno de ellos. La parte de la red puede ser mayor o menor dependiendo del numero de redes con que se desee contar o

del numero de nodos por red."[6]

3.1.4.1 MASCARAS DE SUBRED

Las redes de clase A y B admiten millones o miles de nodos para compartir el mismo identificador de red. IP define lo que se conoce como mascara de subred o simplemente mascara de red como un medio de admitir a tales nodos residir en diferentes redes físicas mientras que mantengan la misma identidad de red.

Cuando a un nodo o a cualquier otro elemento de la red se le asigna una dirección IP, IP deriva su clase de red y dirección de red de esa asignación (148.29). Cuando se requiere depositar un datagrama en un nodo, se compara la dirección de red de la dirección destino enviada por el protocolo de transporte (TCP o UDP) con la dirección que le pertenece. Si la dirección concuerda, entonces IP no mandara el datagrama a un ruteador para ser auxiliado en la entrega, en su lugar, IP asume que el anfitrión se encuentra en la misma red y por tanto trata de entregarla directamente a la dirección designada de red.

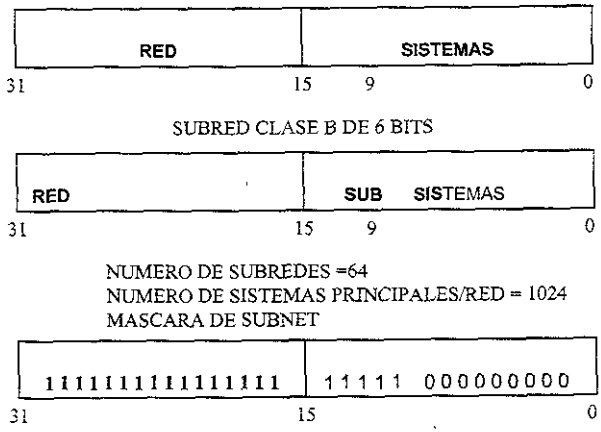

Fig. 3 5 Partes de las dirección IP

[6] Tackett & Gunter, Utilizando Linux 2a. Edicion, Prentice Hall, P. 258.

El establecimiento de una red de subred es una cuestión de determinar donde termina la dirección de la red y donde comienza la dirección del sistema principal. La mascara de la subred contiene todos los 1 en el campo de red y 0 en el campo del sistema principal. Supóngase que una red clase C esta compuesta de lo siguiente:

N= network

H= Nodo

NNNNNNNN.NNNNNNNN.NNNNNNNN.HHHHHHHH

Cada posición representa un solo bit de un espacio de dirección de 32 bits. Si esta red de clase C se dividiera en cuatro redes clase C, el patrón se parecería a lo siguiente:

NNNNNNNN.NNNNNNNN.NNNNNNNN.NNHHHHHH

La mascara de la subred se parecería a lo siguiente:

11111111.11111111.11111111.11000000

Si esta dirección se escribe en notación de puntos base diez, la mascara de la subred seria 255.255.255.192. Esta mascara se utiliza para la comunicación entre nodos en todas las subredes dentro de esta red especifica.

Si se toman tres bits del campo del sistema principal podrán formarse ocho redes y la mascara resultante de red será como se muestra a continuación:

11111111.11111111.11111111.11100000

La mascara de subred es 255.255.255.224. Cada una de las ocho redes debería tener 29 nodos debido a que hay cinco bits de dirección disponibles. (En realidad habría 32 si no fuera porque los números 1,0,127 no son direcciones legales).

Este concepto puede extenderse a las redes de clase B y a las de clase A. La única diferencia es que los campos restantes son 0 (cero)

Considere una red clase B. El espacio de la dirección se divide como sigue:

NNNNNNNN.NNNNNNNN.HHHHHHHH.HHHHHHHH

Si se toman dos bits del campo del sistema principal y se añaden a la parte de red, se utilizara la siguiente mascara de subred. :

11111111.11111111.11000000.00000000

La mascara se escribe como:

255.255.192.0

Los bits necesarios para la mascara de subred pueden tomarse de cualquiera de las posiciones de bit dentro del campo del sistema principal, pero ello conduce a mascaras de subred y exclusiones de direcciones complejas. Por esta razón deberá evitarse en la medida posible.

3.1.5 RUTADO IP

El rutado IP es uno de los métodos más fáciles y eficientes para rutear información en una red compleja. IP distingue entre nodos y puentes. Un puente en TCP/IP es en realidad un ruteador que conecta 2 o más redes con el propósito de proporcionar servicios de envío de datos entre ellas.

La figura 3.6 muestra un puente enviando un datagrama entre dos redes. Un nodo es el sistema final donde se ejecuta la aplicación del usuario. El rutado en nodos esta limitado a la entrega del datagrama directamente al sistema remoto, si ambos nodos se encuentran en

la misma red, si no, IP entrega el datagrama al puente por default. El puente por default esta definido en el nodo durante la configuración de TCP/IP es un ruteador vinculado a la misma red a la que el nodo debe pedir ayuda en entregas hechas a otros nodos en redes remotas.

3.1.5.1 RUTEADORES Y PUENTES

Ruteadores.

Los ruteadores son usados para servicios de rutado entre redes de la misma arquitectura. Cuando este recibe un paquete desde una red, busca la dirección de destino dentro del paquete. Para determinar esto, el ruteador tiene información almacenada dentro de él, que le indica cuales redes están conectadas a él y que redes pueden ser alcanzadas.

Puentes

Un puente tiene funciones similares a la de un ruteador, pero el puente conecta redes que usan diferentes mecanismos de transmisión. Por ejemplo un puente puede conectar redes que usan Ethernet para transmitir paquetes con una que usa fibra óptica. En este caso, el puente debe tener hardware que soporte ambos mecanismos de transmisión.

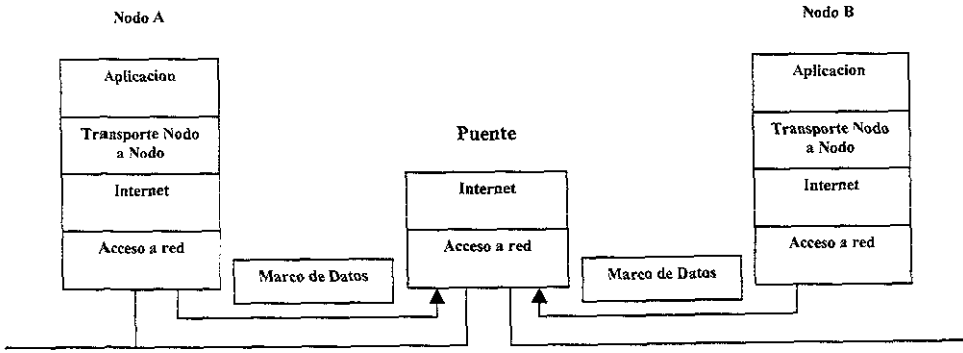

Fig. 3.6 Muestra un puente enviando un datagrama entre dos redes

Un puente puede además conectar redes que usan diferentes protocolos, tal como redes que usan el protocolo TCP/IP con redes que usan el protocolo BITNET, en este caso, el puente tiene software y probablemente hardware que traduce entre los dos protocolos y permite a la información pasar entre las dos redes sin dañarse.

Un router funciona en la capa Red de los protocolos de red. Hay varias formas diferentes de rutar datos. La forma implantada a por una red Internet es el protocolo de información de rutado (RIP)

3.1.5.2 PROTOCOLO DE INFORMACIÓN DE RUTADO (RIP)

El protocolo de rutado funciona de la siguiente manera:

Para mantener una lista de saltos a nodos adyacentes, un router RIP mantiene una tabla de rutado en el router o en la memoria de la computadora. Esta tabla se actualiza a intervalos de 30 segundos con información procedente de routers en la vecindad. La información se utiliza para volver a calcular la ruta de menor coste entre sistemas. Cada router de una red envía, o anuncia, y recibe información de rutado.

El protocolo de rutado esta limitado a la distancia en que se dirige un mensaje a un coste de solo 16. Si el mensaje enviado en un cable cuesta mas de 16, el sistema principal se considera inalcanzable. El coste es un método de asignar valores a las diferentes rutas a través de la red y es una forma de garantizar una ruta eficiente a un destino cuando hay mas de una forma de llegar hasta allí.

Cuando se produce un fallo de red, los routers se ven obligados a volver a aprender las rutas de menor coste. Esto lleva tiempo y puede resultar en la transmisión de mensajes a un coste mayor durante un cierto periodo de tiempo. Cuando un nodo finaliza bruscamente, todos los routers deben de ajustar sus respectivas tablas de rutado. Durante este tiempo pueden perderse mensajes en la red. Después de un cierto periodo de tiempo los routers vuelven a

sincronizarse y el rutado continua.

Las finalizaciones bruscas de los routers también constituyen una fuente de preocupación. En estas circunstancias, los routers adyacentes actualizan su nivel de vecindad a un router averiado en un periodo de 180 segundos. La ruta se elimina de la base de datos del router local si no recibe información de rutado del router caído una vez transcurrido dicho periodo de tiempo.

La información sobre rutado puede obtenerse de varios RFCs. El de mayor interés para los gestores de redes es el RFC 1058, Protocolo de información de rutado (RIP).

El RIP ha sido diseñado para su utilización en redes pequeñas y medianas y esta basado en los protocolos de rutado de Xerox Network systems (XNS). RIP determina la ruta de un mensaje utilizando un algoritmo de rutado de distancia-vector. Este algoritmo asume que cada ruta se le asigna un coste. Este coste puede ser representativo del rendimiento de una red, del tipo de línea o de la deseabilidad de la ruta. El protocolo entonces determina la ruta de menor coste en la que se puede transmitir el mensaje.

RIP no gestiona distancias de rutado, solo costes. Debido a esto, RIP podría no utilizar la ruta física más corta entre dos puntos.

3.1.5.3 SEGMENTACION DE RED

Las redes de Internet están divididas en segmentos por varias razones. Algunas de estas razones están relacionadas a las tecnologías subyacentes de redes. Otras están relacionadas con ubicaciones geográficas. Algunas de las mejores razones para aislar segmentos de red están basadas en la utilización de la red. Si una gran cantidad de trafico de una red tuviera lugar entre unos cuantos nodos, lo mejor seria asilar esos nodos. Este aislamiento disminuye el uso y proporciona a los restantes usuarios una red con un mayor nivel de respuesta.

Otras razones para segmentar son las modificaciones de las tecnologías de red o el establecimiento de comunicaciones entre ellas. Por ejemplo, una zona de oficinas puede estar funcionando bajo una red Token Ring mientras que la zona de fabrica puede estar funcionando bajo una red Ethernet. Cada una de ellas tiene una función diferente. La oficina necesitara de Token Ring para comunicarse con un servidor AS/400. La fabrica puede utilizar Ethernet a fin de permitir la comunicación entre los controladores de fabrica y las computadoras. La información procedente de la fabrica, podría cargarse a la red de la oficina para permitir el seguimiento de pedidos. La conexión entre tecnologías se realiza normalmente a través de routers. Los routers envían solo la información que debe transmitirse desde una red a la otra. Esta información podrá compartirse entre los nodos de las redes respectivas.

El uso excesivo de routers en una red puede llegar a constituir una carga para esta, contrarrestando sus ventajas. La utilización de un router proporciona muy pocos beneficios si todos los nodos de una red deben alcanzar todos los nodos de otra red y viceversa. En este caso, las ventajas del rutado se verían reducidas a causa de la sobrecarga de los protocolos de rutado. En este tipo de situación un puente es una mejor alternativa.

Un puente permite compartir toda la información de las dos redes. El acceso tiene lugar en la capa física en lugar de hacerlo en la capa de red, por lo que no hay ninguna traducción de direcciones ni sobrecarga de rutado. Un puente permite la transmisión de toda la información, incluyendo mensajes de multidifución del sistema. Un router es mejor en aquellos casos en que dos redes comparten información solo ocasionalmente; en caso contrario, el puente seria la elección correcta.

3.1.6 CONFIGURACION DE LAS REDES INTERNET

El diseño y la configuración de una red Internet, comprende muchos tipos de nodos, incluyendo estaciones de trabajo, servidores, impresoras, grandes computadoras, routers, puentes, pasarelas, servidores de impresión y terminales. La red Internet requiere que cada

uno de los dispositivos tenga su dirección IP exclusiva. Un dispositivo puede tener mas de una dirección, dependiendo de su función, pero se requiere por lo menos una dirección para comunicación con los restantes dispositivos.

3.1.6.1 TIPOS DE CONEXIONES

Una red TCP/IP puede constar de varios sistemas conectados a una red de área local o cientos de sistemas con conexiones a miles de sistemas en Internet. Cada organización puede crear el tipo de red que mejor se ajuste a sus necesidades.

La figura 3.7 muestra una red simple. La red consta en varias estaciones de trabajo y un servidor de archivos. A cada estación de la red se le asigna la dirección de red (194.62.23.X). A cada dispositivo se le asigna una dirección individual de nodo. En esta red, hay sitio para conectar impresoras y más estaciones de trabajo a la red. La red no tiene provisión alguna para conexiones a otras redes de área local o amplia.

La red de la figura 3.8 es más compleja. Esta red incluye tres redes independientes interconectadas a través de una combinación de routers y servidores. Cada una de las estaciones de trabajo y ordenadores de cada segmento puede estar o no aislada con lo que respecta a la utilización de información en una de las dos redes restantes. Esta es una característica de la mascara y seguridad de la subred activada en los servidores y routers.

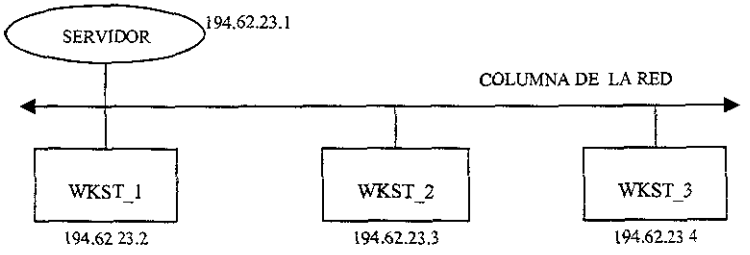

Fig. 3.7 Una red simple

La información de una red se ruta a una de las otras redes según se requiera. El router 1,

entre las redes 1 y 2, proporciona información de rutado entre las dos redes. Si al servidor 1 conectando a las redes 2 y 3 se le ha activado el rutado, la información de la red 3 a la red 2 se ruta. Además, la información puede rutarse desde la red 3 a la red 2 por medio del servidor 1 y desde la red 2 a la red 1 por medio del router 1. El servidor 1, conectando a las redes 2 y 3, tiene dos direcciones IP: una dirección IP en la red 2 y otra en la red 3. Esta es también aplicable al router 1, con direcciones en la red 2 y en la red 1.

Considere una situación en la que hay mucho trafico Internet entre la red 3 y la red 1. En este caso, podría valer la pena ubicar un router adicional entre la red 1 y la red 3. El router adicional puede eliminar algunas de las sobrecargas de rutada sobre el servidor 1 y permitir la transferencia de información entre redes cuando el servidor 1 se encuentre inutilizado.
El router adicional puede añadir a la red un nivel de tolerancia de fallos basada en el hecho de que la información todavía puede rutarse a la red 2 desde la red 3, incluso cuando el servidor 1 se encuentra inutilizado. La ruta entre la red 3 y la red 2 se efectuaría a través de la red 1 y del router 1.

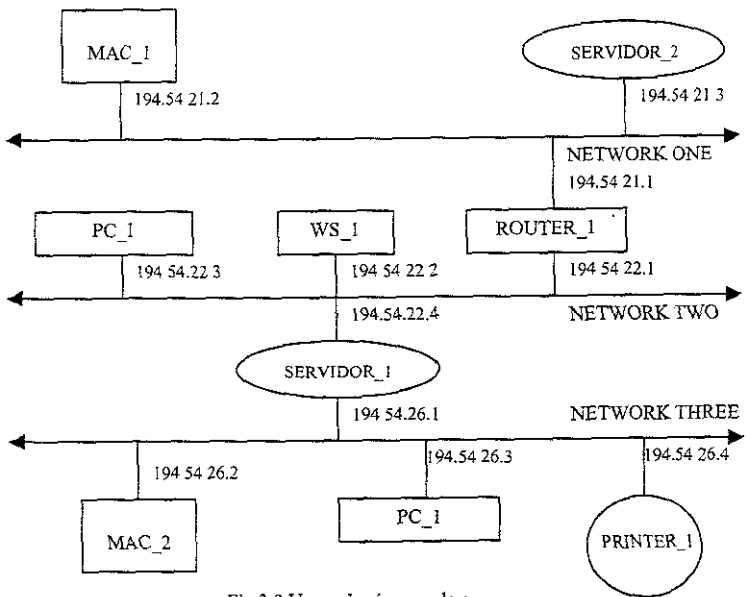

Fig 3.8 Una red más compleja

La tolerancia de fallos en una red mejora su integridad y puede ser particularmente importante en ciertas aplicaciones. Si es necesario compartir información de temporización crítica entre dos redes, deberá proporcionarse una red alternativa entre las redes. Esto podría proporcionarse mediante la utilización de routers adicionales. Es necesario utilizar un parámetro de configuración debido a que estas rutas pueden ser indirectas (a través de una tercera red).

A este parámetro normalmente se le conoce como coste de red. El coste de un salto puede aumentar si aumenta el valor que toma un paquete al cruzar una ruta de red. La ruta preferida por omisión es la ruta de menor coste; la ruta alternativa es la de alto coste. Esta disposición impide que la información se transfiera por rutas de alto coste de forma regular.

Los medios típicamente utilizados por una red Internet pueden consistir en casi cualquier tecnología de red utilizada en la actualidad. El trafico de la red Internet no esta limitado a Ethernet, ARCnet o Token Ring. Puede viajar también en redes RS232 asincronas, líneas T1 y a través de relé de trama.

Se debe tener en cuenta la amplitud de banda que requiere una aplicación. Muchas aplicaciones requieren la transferencia de megabytes de datos, por lo que la amplitud de banda se convierte en una de las consideraciones principales.

Al diseñar la red se debe de considerar el tipo de información a transportar en la red, su ubicación física y la carga de red. Para ayudar a determinar la capacidad de la red deberá de examinar el tipo de estaciones de trabajo, servidor y aplicaciones.

Si se utilizan estaciones de trabajo sin unidades de disco en una red, la red deberá admitir una mayor carga por cada nodo. La razón de esto es que cada estación de trabajo remota, sin unidad de disco, requiere la descarga a través de la red de todo el código correspondiente al sistema operativo. Debido a que todas las aplicaciones y utilidades, así como los archivos de datos, se almacenan remotamente, cada acción realizada en dicha

estación de trabajo requiere un acceso de red.

Otro criterio es la cantidad de trafico NFS que habrá en la red. NFS proporciona servicios remotos de disco virtual, por lo que la información recuperada y almacenada en esos discos remotos es constantemente utilizada en la red.

Otras consideraciones a tener en cuenta son la existencia de grandes imágenes, archivos de intercambio y de pagina utilizados para memoria virtual, aplicaciones de bases de datos distribuidas, trafico de impresión, y trafico entre terminales.

Otros elementos a tener en cuenta son las necesidades de marcado y acceso remoto. Si este acceso esta relacionado al trafico de pantalla y terminal podría bastar con un puerto serie de un sistema ya existente. Si se realiza una conexión de Línea serie IP (SLIP), deberá tener en cuenta que nivel de carga va a tener que soportar la red cuando los usuarios estén cargando utilidades de software, programas y bases de datos a través de líneas telefónicas. Este es un factor a tener muy en cuenta debido a que IP no esta limitado a enlaces de alta velocidad como Novel IPX y otros protocolos de red.

3.2 CONFIGURACION DE UNA RED TCP/IP

La configuración de una red TCP/IP es una de las tareas más comunes con las que cualquier administrador se encontrara en los sistemas Linux.

3.2.1 ARCHIVO DE CONFIGURACION DE TCP/IP

La conexión en red TCP/IP de Linux esta controlada por un conjunto de archivos en el directorio /ect. Estos archivos informan a Linux de su dirección IP, nombre del sistema y nombre de dominio, y controlan las interfaces de red. La tabla 3.3 muestra la función de cada archivo.

Archivo	Descripción
/etc/nodos	Asigna los nombres de sistemas a las direcciones IP.
/etc/networks	Asigna los nombres de dominio a las direcciones de la red
/etc/rc d/rc.inet1	Secuencia que configura y activa sus interfaces Ethernet en el momento del arranque.

Tabla 3 3 Archivos de configuracion de la red TCP/IP de Linux

3.2.1.1 EL ARCHIVO */etc/hosts*

Todas las computadoras en una red TCP/IP tienen una dirección IP, un nombre de sistema canónico y opcionalmente uno o más alias de nombre de sistema. El archivo */etc/hosts* es el método original para asignar nombres de sistema de direcciones IP. "... A continuación se muestra un ejemplo de la red que construyo Tristar Inc. Esta red se compone de una única dirección de red de Clase B asignada a Tristar por el NIC; esta red se ha dividido en dos subredes de clase C. El formato del archivo de sistemas se muestra en el ejemplo siguiente:

```
# /etc/hosts for unix1.tristar.com
#
# For loopbacking.
127.0.0.1       localhost

# This machine
166.82.1.21     unix.tristar.com.unix1          # The local machine

# Other hosts on our network
166.82.1.20     server.tristar.com server       # The server

166.82.1.22     wk1.tristar.com                 # Workstation 1

166.82.1.10     netpr1.tristar.com netpr1       # Networked printer
```

166.82.1.1 gateway.tristar .com gateway # The router

166.82.1.1 gate-if1 # 1st interface on gateway

166.82.2.1 gate-if2 # 2nd interface on gateway

166.82.1.30 unic1t.tristar .com unix1t # Laptop via PLIP

 # end of hosts file." [7]

El formato del archivo hosts se compone de una dirección IP por línea, comenzando en la primera columna, el nombre de sistema canónico asociado con esta dirección y opcionalmente uno o más alias. Los campos están separados por espacios o tabuladores. Las líneas en blanco y el texto que sigue a un carácter # se tratan como comentarios y se ignoran.

La dirección IP 127.0.0.1 se conoce como la dirección de bucle interno local y se reserva para este propósito. Generalmente se le asigna el nombre localhost. Si va a usar su sistema solo, como un sistema autónomo o, va a usar SLIP o PPP para conectarse con el mundo exterior, solo necesita la dirección localhosts en su archivo de sistemas.

La función del archivo */etc/hosts* ha sido asumida en gran parte por el servicio de Nombres de Dominio (DNS) en los sistemas conectados a Internet o a grandes redes internas. Sin embargo, DNS no esta disponible durante el arranque o cuando esta trabajando en modo de usuario único, por lo que es una buena idea colocar la información para los sistemas esenciales, como los servidores y las pasarelas en */etc/hosts.*

En una red con pocos sistemas, que no este conectada a Internet, es mas fácil mantener una lista completa de todos los sistemas en */etc/nodos/* en lugar de configurar y mantener el

[7] Tackett & Gunter. Utilizando Linux 2a edicion, Prentice Hall., P 270)

DNS.

3.2.1.2 EL ARCHIVO */etc/networks*

Al igual que los sistemas, que pueden tener nombres y direcciones IP, Las redes y subredes también pueden denominarse. Esta denominación la maneja el archivo */etc/networks*. "El siguiente es un archivo de ejemplo para Tristar.com.

 # /etc/networks for tristar.com

Localnet 127.0 0.0 # software loopback network

tristar-c1 166.82.1 # Development Group Network, Class C

trstar-c2 166.82.2 #MISNetwork. Class C

 # end of networks file"[8]

El primero es el nombre de localnet y la dirección IP, 127.0.0.0. Si no conecta el sistema Linux a una red TCP/IP o solo usa SLIP o PPP. Esto es todo lo que necesita poner en este archivo. Las líneas siguientes identifican las dos subredes de clase C que Tristar ha hecho a partir de su red de Clase B. Las direcciones IP del archivo de redes incluyen solo la parte de dirección de la red, mas el byte de la subred.

3.2.2 INICIALIZAR INTERFAZ INTERNET

El programa *ifconfig* informa al kernel de Linux de las interfaces de red, como el bucle interno de software y las tarjetas Ethernet. Esto se ha de realizar antes de que Linux pueda

[8] Tackett & Gunter, Utilizando Linux 2a Edicion, Prentice Hall., P 271

usarlos. El programa *ifconfig* también se usa para supervisar y cambiar el estado de las interfaces de red. Una llamada a *ifconfig* seria*ifconfig interfaz dirección*

Así activa la interfaz de red indicada y le asigna una dirección IP. Esto se denomina arrancar una interfaz: La sintaxis de llamada generalizada de *ifconfig* es:

ifconfig interfaz [[- net -nodo] dirección [opciones]]

Los indicadores *-net* y *-nodo* fuerzan a *ifconfig* para que trate la dirección como de red o como de sistema. La tabla 3.4 muestra una lista de los argumentos de la línea de comandos de *ifconfig.*

Normalmente no es necesario usar todas las opciones. *ifconfig* puede ajustar todo lo que se necesita a partir de el nombre de la interfaz, la mascara de la red y la dirección IP asignada. El usuario solo debe de ajustar de forma explícita la mayoría de los parámetros si falla *ifconfig* o la red es compleja.

3.2.2.1 USO DE IFCONFIG PARA INSPECCIONAR LA INTERFAZ DE RED

Al ejecutar *ifconfig* con solo un nombre de interfaz en la línea de comandos se imprime el estado de la interfaz. Al ejecutar *ifconfig* sin argumentos se provoca la salida del estado de todas las interfaces de la red que conoce el kernel. Ejemplo:

"$ ifconfig lo

lo Link encap Local Loopback
inet addr 127.0.0.1 Bcast 127.255.255.255 Mask
255.0 0.0
UP LOOPBACK RUNNNING MTU 2000 Metric 1
RX packets 0 errors 0 dropped 0 overruns 0
TX packets 1658 errors 10 dropped 0 overruns 0

Argumento	Descripción
Interface	El nombre de la interfaz de red. Generalmente es el nombre del controlador de dispositivos seguido de identificación. Este argumento es obligatorio
Aftype	Indica la familia de direcciones que se debería usar para decodificar y mostrar todas las direcciones de protocolo Actualmente se admiten las familias de direcciones inet (TCP/IP) y AX.25 (packet radio amateur). Por omisión es la familia inet.
Up	Al usar esta opción se activa la interfaz indicada
[-] arp	Activa o desactiva el uso del protocolo ARP de la interfaz indicada. El signo menos se usa para desactivar el indicador.
[-] trailers	Activa o desactiva los finales de los marcos Ethernet Actualmente no esta implementado En el sistema de conexión en red de Linux
[-] allmulti	Activa o desactiva el modo promiscuo de la interfaz. Al activar este modo se ordena a la interfaz que envie todo el trafico de la red al kernel, no solo el trafico dirigido a su sistema
Metric N	Ajusta la métrica de la interfaz al valor entero N. El valor métrico representa el "coste" de enviar un paquete por esta ruta Actualmente el kernel Linux no usa el coste de ruta pero se implementara en versiones posteriores.
Mtu N	Ajusta el máximo numero de bytes que puede manejar la interfaz en una transferencia al valor entero N. El código actual de conexión en red en el kernel no maneja la fragmentación IP, por esto, asegúrese de que el valor de MTU es suficientemente grande.
Dstadde addr	Ajusta la direccion IP del otro extremo del enlace punto a punto. Se ha convertido en obsoleto con la palabra clave pontopoint.
Netmask addr	Ajusta la mascara de red IP a la interfaz indicada.
[-] broadcast (aadr)	Ajusta la dirección de emisión de la interfaz cuando se incluye una dirección. Si no se da ninguna dirección, se activa el indicador IFF_BROADCAST de la interfaz indicada. Un signo menos inicial desactiva el indicador.
[-] pointopoint (addr)	Esta opción activa el modo punto a punto en la interfaz indicada. Esto avisa al kernel de que esta interfaz es un enlace directo con otro sistema. La direccion, cuando se incluye, se asigna al sistema en el otro extremo de la lista. Si no se da ninguna dirección, se activa el indicador IFF_pointopoint de la interfaz. Un signo menos inicial desactiva el indicador.
Hw	Esta opción ajusta la dirección de hardware de la interfaz indicada. El nombre de la clase de hardware y el equivalente ASCII de la dirección de hardware han de seguir a esta palabra clave. Actualmente se admiten los Ethernet (ether), AMPR AX 25 (ax25) y PPP (ppp).
Adress	Este es el nombre de sistema o la dirección IP a asignar a la interfaz indicada. Los nombres de sistema usados aquí se solucionan según sus direcciones IP equivalentes. Este parámetro es obligatorio

Tabla 3.4 Argumentos de la línea de comandos de ifconfig.

Este ejemplo usa *lo*, la interfaz de bucle interno de software. El usuario puede ver la dirección IP asignada inet addr, la dirección de emisión Bcast y la mascara de red Mask. La interfaz es UP con un MTU de 2000 y un Metric de 1. Las dos ultimas líneas dan estadísticas sobre el numero de paquetes recibidos Rx y transmitidos TX, junto con las cuentas de errores de paquete, anulaciones y desbordamientos."[9]

3.2.2.2 CONFIGURACION DE LA INTERFAZ DE BUCLE INTERNO DE SOFTWARE

Todos los sistemas Linux que tengan un nivel de conexión en red instalado en el kernel tienen una interfaz de bucle interno de software. Esta interfaz se usa para comprobar las aplicaciones de conexión en red y para suministrar una red para servicios TCP/IP locales, como el INN cuando el sistema no esta conectado a la red real.

El nombre de la interfaz de red del sistema bucle interno es *' lo '*. ejemplo:

ifconfig lo 127.0.0.1 <Intro>

Esta instrucción activa la interfaz de bucle interno y le asigna la dirección 127.0.0.1, que es la dirección que se usa tradicionalmente para el bucle interno porque la red de Clase A 127.0.0.0, nunca será asignada a nadie por el NIC.

Para que el sistema de bucle interno sea totalmente operacional, el usuario debe de añadirle una ruta con el comando *route*

3.2.2.3 CONFIGURACION DE UNA INTERFAZ DE RED

La configuración de una interfaz de red Ethernet requiere un poco mas de trabajo,

[9] Tackett & Gunter, Utilizando Linux 2a Edicion, Prentice Hall., P 274

especialmente si usa subredes, La llamada básica a *ifconfig* es la siguiente para unix.trstar.com:

ifconfig eth0 unix1

"Esto provoca que *ifconfig* active la interfaz Ethernet 0, mire la dirección IP de unix1 en el archivo */etc/hosts* y lo asigne a esta interfaz. Al examinar la interfaz eth0 en este punto se ve el siguiente código:

$ ifconfig eth0
eth0 link encap 10Mbps Ethernet HWaddr 00:00:E1:54:3B:82
inet addr 166.82.1.21Bcast166.82.1.255 Mask
255 255.255.0
UP BROADCATS RUNNING MTU 1500 Metric0
RX packets 3136 errors 217 dropped 7 overrun 26
TX packets 1752 errors 25 dropped 0 overrun 0"[10]

Debe observarse que *ifconfig* ajusto automáticamente la dirección de emisión y la mascara de red basándose en la dirección IP que encontró en */etc/hosts*. Si el usuario usa subredes, ha de indicar explícitamente la dirección de emisión y la mascara de red. Por ejemplo, si se tiene una red de Clase C y se usa el primer bit en la parte del sistema de la dirección para hacer dos subredes, se debe indicar la dirección de emisión y la mascara de red cuando se ejecute *ifconfig.*

ifconfig eth0 unix1 broadcast 166.82.1.127 netmask
255.255.255.128

[10] Tackett & Gunter, Utilizando Linux 2a Edicion, Prentice Hall., P 275

3.2.2.4 CONFIGURACION DE INTERFACES IP PARALELAS

SLIP y PPP

El kernel de Linux admite dos protocolos de línea serie para la transmisión de trafico de protocolo Internet IP, SLIP (Serial Line Internet Protocol) y PPP (Point to Point Protocol). Fueron desarrollados como una alternativa "económica" a las caras instalaciones de línea alquiladas para conectarse a Internet. Cualquier persona con un módem de velocidad razonablemente alta y un proveedor de servicios que soporte estos protocolos puede conectar por IP su sistema Linux por un costo muy bajo comparado con los sistemas con líneas alquiladas.

Los controladores SLIP para Linux estuvieron disponibles poco después de que este se publicara por primera vez. El soporte PPP se ha agregado desde hace poco pero es muy estable.

Las interfaces paralelo IP (PLIP), de línea en serie IP (SLIP) y de protocolo punto a punto (PPP) las gestiona *ifconfig* de forma diferente. Para arrancar una interfaz PLIP, se añade la opción pointopoint en la línea de comandos de *ifconfig*. Suponga que la computadora portátil de Tristar unix1t esta conectada al primer puerto paralelo en unix1. El usuario llama *ifconfig* de la forma siguiente para activar el enlace PLIP:

ifconfig plip0 unix1 pointopoint unix1t

Así se activa la interfaz Plip0 con la dirección IP de unix1, se ajusta el indicador de pointopoint y se avisa a la interfaz de que la dirección IP del otro extremo del enlace es unix1t. *ifconfig* busca en las direcciones IP unix1 y unix1t en /etc/hosts y asigna apropiadamente las direcciones. En una computadora portátil el usuario usa la llamada análoga

ifconfig Plip0 unix1t pointopoint unix1

3.2.3 RUTADO DE TCP/IP

El rutado determina la ruta de acceso que toma un paquete desde su fuente, a través de la red, hasta su destino. Esta ruta se determina comparando la dirección IP, de destino con las tablas de rutado del kernel y transmitiendo el paquete al sistema indicado, que puede, o no , ser el destino del paquete. La tabla de rutado del kernel contiene información del tipo " Para ir a la red X desde el sistema Y, mande el paquete al sistema Z con un coste 1", junto con los valores de tiempo de expiración y fiabilidad de esta ruta.

3.2.3.1 POLITICA DE RUTADO

El primer paso para instalar un rutado en la red es decidir una política de rutado. En el caso de las redes pequeñas y no conectadas, es suficiente usar el comando route para instalar rutas estáticas en cada sistema en el momento del arranque. Las grandes redes con numerosas subredes o redes conectadas a Internet requieren el uso de un rutado dinámico. El programa de rutado suministra un rutado dinámico al comunicarse con programas de rutado de otros sistemas e instalar rutas basadas en lo que aprende sobre la topología de red.

Una estrategia muy común combina rutado estático y dinámico. Los sistemas de cada subred usan rutado estático por los paquetes que no concuerdan con ninguna otra ruta de la tabla de rutado, esta definida en un sistema pasarela que realiza rutado dinámico y que tiene información sobre el resto del mundo. De esta forma se pueden construir grandes redes, minimizando la complejidad de los archivos de configuración y el ancho de banda usado por los programas de rutado dinámicos.

3.2.3.2 USO DEL PROGRAMA */sbin/route*

El programa */sbin/route* maneja la tabla de rutado del kernel y se usa para definir rutas estáticas a otras computadoras o redes a través de interfaces que han sido configuradas y activadas por *ifconfig*. Esto se realiza normalmente en el momento del arranque con la secuencia */etc/rc.d/rc.inet1*. La tabla 3.5 describe los argumentos de la línea de comandos de */sbin/route*.

Argumentos	Descripción
[none]	Al no dar ninguna opción a /sbin/route se provoca la salida de la tabla de rutado actual
- n	Provoca la misma salida al argumento anterior, pero substituye los nombres del sistema por sus direcciones IP numéricas.
del	El argumento 'del' suprime la ruta de la dirección de destino indicada, de la tabla de rutado.
add	El argumento 'add' agrega a la tabla de rutado una ruta a la dirección o red indicadas.

Tabla 3.5 Argumentos de la línea de comandos de sbin/route

Examen de la tabla de rutado del kernel. Ejecutar */sbin/route* sin ningún argumento de línea de comandos o solo -n provoca la salida de la tabla de rutado siguiente:

/sbin/route

Kernel routing table

Destination	*Gateway*	*Genmask*	*Flags*	*Metric*	*Ref*	*Uselface*
127.0.0 0	*	*255.0.0.*	*U*	*0*	*0*	*100lo*

La tabla 3.6 muestra la explicación de los campos del informe de rutado

"A continuación se presenta el ejemplo del unix1t, La computadora portátil de la red Tristar, con un enlace SLIP hacia arriba y ejecutándose:

Campo	Descripción
Destination	El destino de la dirección IP de la ruta
Gateway	El nombre del sistema o de la dirección IP de la pasarela que usa la ruta. Si no existe ninguna pasarela se imprime un carácter
Genmask	La mascara de red de la ruta. El kernel la usa para definir la generalidad de una ruta realizando AND bit a bit entre la Genmask y la dirección IP del paquete, antes de compararlo con la dirección IP de destino de la ruta
Flags	Los indicadores de la ruta. (U=arriba, H=sistema, G=pasarela, D=ruta dinámica, M=Modificada)
Metric	El coste métrico de la ruta. Actualmente no esta admitido en el nivel de conexión en red del kernel.
Ref	El numero de otras rutas que dependen de la presencia de esta ruta.
Use	El numero de veces que se ha usado la entrada a la tabla de rutado
Iface	La interfaz de red a la que la ruta suministra paquetes

Tabla 3.6 Explicación de los campos del informe de la tabla de rutado

/sbin/route

Kernel routing table

Destination	*Gateway*	*Genmask*	*Flags*	*Met ric*	*Re f*	*Use*	*Iface*
Slip.tristar.c	*	*255.255.255.255*	*UH*	*0*	*0*	*0*	*s10*

127.0.0.0	*	*255 0.0.0*	*U*	*0*	*0*	*100*	*lo*
dafault	*slip.tristar.c*	*	*UG*	*0*	*0*	*1*	*s10"*

[11]

Todo sistema conectado a una red ha de tener una ruta por omisión en su tabla de rutado. La ruta por omisión se usa cuando ninguna otra entrada a la tabla concuerda con el destino del paquete.

La entrada a la tabla del bucle interno es la misma que antes y presenta dos nuevas entradas. La primera indica una ruta a slip.tristar.com, la segunda una ruta por omisión que usa slip.tristar.com como pasarela.

Agregación de rutas estáticas. El ususario agrega rutas a la tabla de rutado ejecutando el programa de rutado con el argumento add. La sintaxis de los argumentos de la línea de comandos es

route add [-net / -nodos] addr [gw gateway] [metric cost] [netmask mask] [dev device]

La tabla3.7 describe los argumentos de la línea de comandos que usa el comando *route add*

Cuando agregue una ruta de pasarela a la tabla de rutado, hay que asegurarse de que la pasarela indicada es accesible. Normalmente se tendrá que agregar una ruta estática para la pasarela antes de añadir la ruta que usa esta.

A continuación se muestran algunos ejemplos, primero la interfaz de bucle interno. Después de configurarlo con ifconfig, se debe de agregar la ruta como se muestra a continuación

[11] Tackett & Gunter, Utilizando Linux 2a Edicion, Prentice Hall., P 277

Argumento	Descripción
- net / - nodo	Trata la dirección indicada como dirección de red o de sistema
Ha	La dirección destino de la nueva ruta. Esta puede ser una dirección IP, un nombre de sistema o de red
gw Ha	Indica que cualquier paquete para esta dirección sea rutado a través de la pasarela indicada.
metric Ha	Esta opción aun no esta implementada
netmask Ha	Indica la mascara de red de la ruta que se agrega. El programa de rutado supondrá lo que en circunstancias normales no es necesario que lo indique.
dev HA	Fuerza la acción de la nueva ruta con el dispositivo de interfaz de red indicado. También aquí la ruta supone, normalmente de forma correcta, que dispositivo usar para la nueva ruta, por lo que no tendrá que usarlo demasiado.

Tabla 3.7 Argumentos de la línea de comandos que usa el comando route add.

route 127.0.0.1

No se necesita mas porque *route* compara la dirección recibida con las direcciones de las interfaces conocidas y asigna la interfaz de bucle interno a la nueva ruta. El siguiente ejemplo muestra como definir el rutado para el enlace SLIP en el sistema unix1t Tristar después de que este se ha establecido e *ifconfig* se ha usado para activar la interfaz:

route add slip.tristar.com
route add default gw slip.tristar.com

El primer comando agrega una ruta estática para el sistema slip.tristar.com y el segundo informa al kernel de que se use el slip.tristar.com como una pasarela para todos los paquetes con destinos desconocidos.

Hay que asegurar que cualquier nombre del sistema se use en el comando *route* este en el archivo */etc/hosts*, de forma que *route* pueda encontrar la dirección IP del nombre; en caso contrario *route* fallara.

Si se construyen subredes en la red, partiendo de la dirección IP por la mitad de un octeto, se tendrá que indicar la mascara de red requerida cuando se ejecute *route*. Por ejemplo, si se tiene una red de Clase C y cuatro subredes que usan los dos primeros bits del ultimo octeto, se debe de ejecutar *route* de la siguiente forma:

route add nodoname netmask 255.255.255.192

Esto garantiza que *route* ponga la mascara de red correcta en la entrada de la tabla de rutado.

En el caso de Ethernet y otras interfaces de red de emisión, se deben de agregar rutas que informen al kernel de que red esta accesible por medio de cada interfaz configurada. Después de haber usado *ifconfig* para arrancar la interfaz de red eth0 en unix1.tristar.com tal como se hizo antes, se debe ejecutar *route* para instalar la ruta en la red de esta interfaz.

route add -net 166.82.1.0

Esto puede parecer insuficiente para definir correctamente la entrada a la tabla de rutado; no se indica ninguna interfaz, *route* lo maneja comparando la dirección IP de la línea de comandos con la dirección IP de cada interfaz de red. Asigna la ruta a la interfaz que concuerda con él. En este caso se ha asignado la dirección 166.82.1.21 a eth0 con una mascara de red 255.255.255.0, que concuerda con la dirección de red dada en el comando *route*. De esta forma *route* instala una ruta en la red 166.82.1.0 usando la interfaz eth0, tal como se muestra a continuación:

$ route
kernel routing table

Destination	*Gateway*	*Genmask*	*Flags*	*Metric*	*Ref*	*Uselface*
166.82.1.0	*	*255.255.255.0*	*UN*	*0*	*0*	*0 eth0*
127.0.0.0	*	*255.0.0 0*	*U*	*0*	*0*	*100 10*

Para informar a unix1 de como se alcanza la otra subred, necesita como seguridad dos entradas mas de tabla de rutado:

route add gateway.tristar.com
route add net 166.82.2.0 gw gateway.tristar.com

Esto añade una ruta estática a gateway.tristar.com y añade una ruta de red para 166.82.2.0 usando el gateway.tristar.com como pasarela para la red, tal como se muestra a continuación:

$ route
Kernel routing table

Destination	*Gateway*	*Genmask*	*Flags*	*Metric*	*Ref*	*Uselface*
gateway.tristar	*	*255.255.255.0*	*UH*	*0*	*0*	*0 eth0*
166.82.1.0	*	*255.255.255.0*	*UN*	*0*	*0*	*0 eth0*
166.82.2.0	*gateway.tristar*	*255.255.255.0*	*UN*	*0*	*0*	*0 eth0*
127.0.0.0	*	*255.0.0.0*	*U*	*0*	*0*	*100 lo*

Esto muestra la ruta estática que se añadio para gateway.tristar.com y la ruta con pasarela a la red 166.82.2.0.

3.2.4 ESCRITURA DE LA SECUENCIA DE ARRANQUE /etc/rc.d/rc.inet 1

Esta secuencia se llama desde */etc/rc.d/rc.M* cuando Linux cambia a modo multiusuario. Ha

de poner en el archivo *rc.inet1* los comandos que inicializan sus interfaces de red y definir todas las rutas estáticas.

"Si el sistema es autónomo, solo se tiene que inicializar la interfaz de bucle interno. A continuación se muestra como se puede hacer:

```
#! /bin/sh
#
#rc.inet1        Initicializa el sistema de red
#Activate the loopback device
/sbin/ifconfig lo 127.0.0.1
/sbin/route add -net 127.0.0.0"
```

El siguiente es más complejo. Inicializa el dispositivo de bucle interno, una interfaz Ethernet y define el rutado de una pasarela al resto del mundo:

```
# ! /bin/sh
#
#rc.inet1        Inicializa el sistema de red
#Activate the loopback device
/sbin/ifconfig 10 127.0.01
/sbin/route add -net 127.0.0.0
# Initialize Ethernet interface 0
/sbin/ifconfig eth0 166.82.1.21 netmask 255.255.255.0
[ccc] broadcast 166.82.1.255
/sbin/route add -net 166.82.1.0 netmask 255.255.255.0
# Now, add route to gateway machine and to rest of world
/sbin/route add -host 166.82.1.1  # Route  to gateway machine
/sbin/route add default gw 166.82.1.1   # Route to rest of world.[12]
```

[12] Tackett & Gunter, Utilizando Linux 2a Edicion, Prentice Hall., PP 280,281

Esta secuencia inicializa el dispositivo de bucle, configura la interfaz 0 para que use la dirección IP 166.82.1.21 y añade una ruta a través de eth0 a la red 166.82.1.0. Los dos últimos comandos de rutado definen una ruta estática al sistema pasarela en la dirección IP 166.82.1.1 y definen la ruta predeterminada a dicho sistema. Cualquier paquete que este destinado a direcciones que no estén en la red 166.82.1.0 se enviaran a la pasarela, que ha de saber como tratarlas.

3.2.5 SUPERVISION DE UNA RED TCP/IP

El programa *netstat* es una herramienta muy valiosa para la supervisión de la red TCP/IP. Puede mostrar la tabla de rutado del kernel, el estado de red. La tabla 3.8 describe los argumentos de la línea de comandos de *netstat*.

Argumento	Descripción
-a	Muestra información sobre todas las conexiones Internet, incluyendo las que solo están escuchando.
-i	Muestra estadísticas sobre todos los dispositivos de la red.
-c	Muestra continuamente el estado actualizado de la red. Esto hace que netstat liste el estado de la red una vez por segundo hasta que se interrumpa
-n	Muestra las direcciones remotas y locales e información del puerto en forma numérica, en lugar de solucionar los nombres de sistema y los nombres de servicio.
-o	Muestra el tiempo de expiración del estado de temporizador y del estado de intento de cada conexión de la red
-r	Muestra la tabla de rutado del kernel
-t	Muestra solo información del socket TCP. Incluye a los que solo están escuchando.
-u	Solo muestra información del socket UDP.
-v	Muestra la información de la versión de netstat.
-w	Muestra información sin procesar del socket
-x	Muestra información del socket de dominio UNIX

Tabla 3.8 Argumentos de la línea de comandos del programa netstat

A continuación se describen cada una de las funciones:

3.2.5.1 REPRESENTACIÓN DE LAS CONEXIONES DE RED ACTIVAS

Al ejecutar *netstat* sin argumentos en la línea de comandos se genera un listado de las conexiones de red activas del sistema. A continuación se muestra la salida predeterminada de *netstat* :

netstat <intro>

Active Internet Connections

Proto	*Recv-Q*	*Send-Q*	*Local Addres*	*Foreign Addres*
➔	*(state)*	*User*		
TCP		*0*		*unix1.tristar.com:1266*
				server. tristar.:telnet
➔	*ESTABLISHED*	*lance*		

Active UNIX domain sockets

Proto RefCnt Flags	*Type*	*State*	*Path*

Supervisión de una red TCP/IP netstat

Unix	*[ACC]*	*SOCK_STREAM*	*LISTENING*	*/dev/*
printer				
Unix 2	*[ ]*	*SOCK_STREAM*	*CONECTED*	*/dev/log*
Unix 2	*[ ]*	*SOCK_STREAM*	*CONECTED*	
Unix 1	*[ACC]*	*SOCK_STREAM*	*LISTENING*	*/dev/log*

La primera sección muestra una conexión de protocolo TCP activa desde el puerto 1266 en unix1.tristar.com hasta el puerto Telnet en server.tristar.com con el usuario *lance*. La Tabla 3.9 describe los campos en el listado de conexiones activas de Internet.

La segunda sección muestra los sockets de dominio UNIX activos. La sockets de dominio UNIX son un mecanismo IPC que usa el sistema de archivos UNIX como punto de reunión. Los procesos crean archivos especiales en el sistema de archivos que son abiertos por otros procesos del sistema que quiere comunicarse. Las líneas de código anteriores muestran dos sockets a la escucha, uno en */dev/printer* y el otro en */dev/log*. También hay dos sockets conectados, uno a */dev/log* y otro sin ruta de acceso asociada. La Tabla 3.10 describe los campos del listado de sockets de dominio UNIX activos.

Al llamar a *netstat* con la opción *-o* agrega la información del estado interno al listado de conexiones de Internet activas. A continuación se muestra un ejemplo de esto:

$ netstat -0 <Intro>

Active Internet connections

Protp Recv-Q Local Address Foreign Address (State)

user

tcp 0 0 localnodo:1121 localnodo:telnet ESTABLISHED

lance off (0.00/0)

tcp 0 0 Localnodo:telnet localnodo:1121 ESTABLISHED

root on (673.69/0)

Campo	Descripción
Proto	El protocolo usado por esta conexión, TCP o UDP.
Recv-Q	El numero de bytes recibidos en este socket, pero no copiados aun por el programa de usuario.
Send-Q	El numero de bytes enviados al sistema remoto, que no han sido reconocidos.
Local address	Nombre del sistema local y numero de puerto asignados a esta conexión. La dirección IP del socket se resuelve según el nombre de sistema canónico para esta dirección y se traduce el numero de puerto al nombre del servicio, a menos que se use el indicador -n.
Foreign address	El nombre del sistema exterior y el numero de puerto asignado a esta conexión El indicador -n afecta a este campo al igual que el campo Local address.
ESTABLISHED	La conexión esta totalmente establecida
SYN_SENT	El socket esta actualmente intentando conectarse al sistema remoto
SYN_RECV	Se inicializa la conexión
FIN_WAIT1	Se ha cerrado el socket y esta esperando la conexión para desconectar
FIN_WAIT2	Se ha cerrado la conexión. El socket esta esperando la desconexion desde el sistema remoto
TIME_WAIT	El socket esta cerrado y espera una retransmisión de desconexión del sistema remoto.
CLOSED	El socket no esta funcionando
CLOSE_WAIT	El sistema remoto ha desconectado su conexión. El sistema local esta esperando que el socket se cierre.
LAST_ACK	La conexión remota esta desconectada y el socket esta cerrado. El sistema local esta esperando.
LISTEN	El socket esta escuchando, a la espera del intento de conexión de entrada.
UNKNOWN	Es estado del socket no es conocido
Usuario	La ID de conexión al sistema del usuario propietario del socket

Tabla 3.9 Campos de conexiones activas de Internet

Campo		Descripción
Proto		El protocolo en uso en este socket. Generalmente será Unix.
RefCnt		El numero de procesos conectados a este socket.
Flags		Los indicadores para este socket. Actualmente, el único indicador conocido es SO_ACCEPTON (ACC), que indica que el socket esta desconectado y que el proceso que hizo el socket esta esperando una petición de conexión.
Type		El modo como se accede al socket. Este campo contendrá una de las siguientes palabras clave:
	SOCK_DGRAM	Datagrama, modo sin conexión.
	SOCK_STREAM	Modo de flujo orientado a la conexión
	SOCK_RAW	Modo sin procesar.
	SOCK_RDM	Modo de mensaje entregado de forma fiable
	SOCK_SEQPACKET	Modo de paquete secuencial
	UNKNOWN	Modo no conocido por el programa netstat
State		El estado actual del socket. Se usan las siguientes palabras clave:
	FREE	El socket no esta asignado
	LISTENING	El socket esta esperando una petición de conexión.
	UNCONNECTED	No hay actualmente una conexión con el socket
	CONNECTING	El socket esta intentando establecer una conexion.
	CONNECTED	El socket esta actualmente conectado
	DISCONNECTING	El socket esta intentando desconectar una conexión.
	UNKNOWN	El estado del socket es desconocido. No lo vera bajo condiciones de operación normales.
Path		Este es el nombre de la ruta de acceso usado por otros procesos para conectarse al socket.

Tabla 3.10 Campos de listado de sockets de dominio UNIX activos.

Los datos agregados están al final de cada línea e incluyen un contador de bytes de

retransmisión del receptor, un contador de bytes de retransmisión del transmisor, el estado (on/off) y los valores (tiempo/intento) del temporizador. El tiempo que se muestra es el que queda antes de que expire el temporizador. El intento es la cuenta de reintentos de la transmisión de datos actual. Estos datos son útiles en el diagnostico de problemas de red al fácilitar la visualizacion de que conexión tiene problemas.

3.3 Configuración del Servicio de Nombres Dominio (DNS)

3.3.1 TABLA DE NOMBRES Y DE NODOS

El método más simple para resolver los nombres de nodo de las direcciones IP involucra el mantenimiento de una tabla de nodos en cada sistema UNIX. Esta tabla se encuentra normalmente en el archivo etc/host. Esta compuesta de una base de datos en el que cada entrada describe la dirección IP de un nodo y su nombre asociado (o asignado). A continuación se presenta un ejemplo de un archivo de host.

#

#Database if IP addresses and host names (Including aliases) the format of

#this files is as follows:

#IP addres	*Hostname*	*aliases*
127.0.0.1	*localhost*	
100.0.0.1	*netrix.unicom.com*	*Netrix*
100.0.0.2	*jade.ott.unicom.com*	*Jade*
100.0.0.2	*orbit.ott.unicom.com*	*Orbit*
198.53.237.1	*pixel.ny.unicom.com*	*Pixel*
198.53.237.20	*emerald.ny.unicom.com*	*emerald*

Como se muestra, las líneas de texto precedidas por el signo # son comentarios. La cuarta

línea de comentario explica el formato de las entradas del archivo. Cada entrada esta compuesta por la dirección IP del nodo, el nombre del dominio completo y su alias opcional para simplificar su referencia.

Para trasladar el alias o el nombre de dominio a la dirección IP, todas las aplicaciones TCP/IP tales como Telnet y FTP tienen un mecanismo que busca en la tabla de nodos y entrega la dirección IP correspondiente.

El origen de las tablas */etc/host* se remonta a la época de los inicios de Internet, cuando este estaba compuesto de pocos nodos, haciendo la asignación de nombres a las direcciones una tarea trivial. El centro Nacional de Información, mantenía una base de datos llamada host.txt, la cual contenía información (incluyendo los nombres de maquina y las direcciones IP) pertinente a estos nodos. Para conectar otros nodos a Internet, se le tenia que asignar un nombre que no causara conflictos con los que ya existían. El nombre, la dirección IP y otra información del nodo eran incluidas en la base de datos.

Cada sistema conectado a Internet, tenia que mantener archivos derivados del archivo hosts.txt, llamados */etc/hosts* y */etc/networks.* Mientras que el primer archivo contiene las direcciones IP para la asociación de nombres, el ultimo archivo contiene las direcciones IP para la asociación del nombre de red. Al crecer Internet, este esquema presento las siguientes fallas:

- Colisión de nombres. Con un espacio de nombre plano, la colisión de nombres aumento con el numero de nodos conectados. Esto hizo más difícil la asignación de nombres.

- Administración de nombres. Ya que cada nuevo nombre tenia que ser aprobado por la autoridad central en el NIC, la tarea de administración de nombres se volvió muy demandante.

- Consistencia asegurada. Como el numero de sitios conectados creció en tamaño, se

volvió mas caro y difícil asegurar las consistencias de copias de la tabla de nodo a cada sitio.

- Incremento del trafico de la red. Como tanto las bases de datos como los sitios crecieron en tamaño, el trafico generado en Internet de peticiones para actualizar la base de datos se volvió muy alto.

Para remediar estas fallas, se requirió de un nuevo sistema para la resolución de nombres. El sistema debía soportar un espacio de nombre que permitiera la descentralización de su administración. Este objetivo puede ser alcanzado si el espacio de nombre propuesto, puede seguir una jerarquía que permita la partición del espacio de nombre en pequeñas partes, o dominios. Su administración por lo tanto puede ser delegada a otras organizaciones en Internet.

El sistema de nombres de dominio (RFC 1035) es una base de datos jerárquica, distribuida de información referente a los nodos en la red. La jerarquía permite la subdivisión del espacio de nombre en partes administrables independientemente llamadas dominios o subdominios. Al ser distribuida, permite el cambio de lugar del subdominio en nombres de servidores que pertenecen al sitio al que es delegada la administración del subdominio. Tales servidores son llamados servidores de nombre autoritarios.

La asignación de los nombres de sistema Internet a las direcciones IP es una tarea que requiere mucha consideración. Con el crecimiento explosivo de Internet en los últimos años, el sistema original de mantener las asignaciones de los nombres de sistema con sus direcciones IP en un archivo ASCII plano local pronto demostró no ser practico. Con miles de computadoras en la Red, y más que se añaden cada día, se necesitaba un nuevo sistema. Este nuevo sistema fue una base de datos distribuida (que abarcaba toda la red), conocida como BIND.

Este sistema, también conocido como Servicio de Nombres de Dominio, Sistema de

Nombres de Dominio o DNS, suministra un nombre de sistema efectivo, relativamente transparente al mecanismo de asignación de direcciones IP. DNS es notoriamente difícil de configurar, pero una vez que se ha logrado es bastante fácil de mantener.

3.3.2 INTRODUCCIÓN A DNS

DNS suministra un mecanismo para la conversión de direcciones IP a nombres mnemónicos que representan sistemas, redes y alias de correos. Lo hace dividiendo todo el espacio de nombres y de IP de Internet en diferentes grupos lógicos. Cada uno de estos grupos tiene autoridad sobre sus propias computadoras y otra información.

Debido a que DNS es un tema complicado, tiene su propio conjunto de términos especializados. La Tabla 3.11 da las definiciones de algunos términos de DNS usados comúnmente.

Al principio, cuando se formo por primera vez Internet, el numero de sistemas en la Red era muy pequeño. Era bastante fácil mantener la asignación nombre/dirección. Cada sistema simplemente tenia una lista completa de todos los nombres de sistemas y direcciones en un archivo local. Al acelerarse el crecimiento de Internet el sistema se volvió rápidamente poco manejable. Cuando se agregaba un sistema era necesario actualizar todos los archivos de sistemas. El tamaño de estos archivos comenzó a crecer demasiado. Claramente se necesitaba otra solución, y esta fue el Sistema de Nombres de Dominio.

DNS se dividió conceptualmente en las tres partes siguientes:

- Espacio del nombre de dominio
- Servidores de nombre

- Agentes de resolución

Termino	Definición
Dominio	La enidad lógica u organización que representa una parte de una red. Por ejemplo, **unc.edu** es el nombre del dominio primario de la Universidad de North Carolina en Chapel Hill.
Nombre de dominio	La parte de un nombre de sistema que representa el dominio que contiene el sistema. Por ejemplo, en la dirección **sunsite.unc.edu**, el nombre de dominio es **unc.edu** se usa alternativamente con dominio.
Sistema	Una computadora en una red.
Nodo	Una computadora en una red.
Servidor de nombres	Una computadora que proporciona los servicios DNS para asignar nombres DNS a direcciones IP.
Solucionar	La acción de traducir un nombre DNS a su dirección IP correspondiente.
Agente de resolución	Un programa o rutina de biblioteca que extrae la información DNS de un servidor de nombres.
Solución inversa	Concordancia de una dirección IP dada con su nombre DNS.
Falsificación	La acción de aparecer en la red como si tuviese una dirección IP o un nombre de dominio diferentes.

Tabla 3 11 Terminos DNS usados comunmente.

El espacio del nombre de dominio es una especificación de una estructura en árbol que identifica un conjunto de sistemas y suministra información sobre ellos.

Conceptualmente, cada nodo en el árbol tiene una base de datos de información sobre los sistemas bajo su autoridad. Las consultas tratan de extraer la información apropiada de esta base de datos. De forma simple, es el listado de todos los diferentes tipos de información, nombres, direcciones IP, alias de correos, etc., que se pueden consultar en el sistema DNS.

Los programas que guardan y mantienen los datos localizados en el espacio de nombres de dominio se conocen como servidores de nombres. Cada uno de ellos tiene una información completa sobre un subconjunto de espacio de nombres de dominio y tiene información en antememoria sobre otras partes. Un servidor de nombres tiene información completa de su área de autoridad. Esta información autorizada se divide en áreas conocidas como zonas, que se pueden dividir entre diversos servidores de nombres para suministrar un servicio redundante a una zona. Cada servidor de nombres conoce otros servidores de nombres que son responsables de diferentes zonas. Si entra una petición de información de una zona de la que es responsable un servidor de nombres determinado, el servidor de nombres simplemente devuelve la información. Sin embargo, si entra una petición en busca de información de una zona distinta, el servidor de nombres contacta con el servidor apropiado con autoridad sobre tal zona.

Los agentes de resolución son simplemente programas o rutinas de biblioteca que extraen información de servidores de nombres en respuesta a una consulta sobre un sistema en el espacio de nombres de dominio.

3.3.3 EL AGENTE DE RESOLUCION

El primer paso para usar DNS es configurar la biblioteca del agente de resolución de su computadora. Si se quiere usar la resolución de nombres DNS se debe configurar el agente de resolución local, incluso si no va a ejecutar un servidor de nombres de dominio local.

3.3.3.1 EL ARCHIVO */etc/host.conf*

Las bibliotecas del agente de resolución local están configuradas a través de un archivo denominado *host.conf* que esta situado en el directorio */etc*. Este archivo informa al agente de resolución sobre que servicios usar y en que orden. Este archivo es un archivo ASCII que tiene una lista de las opciones del agente de resolución, una por línea. Los campos de

este archivo pueden estar separados por espacios o tabuladores. El carácter # indica el comienzo de un comentario. Se pueden indicar varias opciones en el archivo host.conf. La Tabla 3.12 da una lista de estas opciones.

A continuación se muestra un ejemplo de un archivo de configuración */etc/host.conf* que usa estas opciones:

 # Sample /etc/host.conf file

 # Lookup names via DNS first then fall back to /etc/hosts

order bind host

 # We don't have machines with multiple addresses

multi off

 # check for IP address spoofing
nospoof on

 # and warn us if someone attempts to spoof
alert on

 # Trim the tristar.com domain name for nodo lookups
trim tristar.com

Este ejemplo muestra una configuración general de agente de resolución para el dominio

Opción	Descripción
Order	indica en que orden se prueban los distintos mecanismos de resolución de nombres. Los servicios de resolucion indicados se. prueban en el orden listado Se admiten los siguientes mecanismos de resolución de nombres:
Hosts	Se intenta solucionar el nombre mirando en el archivo /etc/host local.
Bind	Consultar un servidor de nombres DNS para solucionar el nombre.
Nis	Usar el protocolo de Servicio de Información de Red (NIS) para intentar solucionar el nombre del sistema
Alert	Toma off u on como argumentos. Si se activa on. cualquier intento de falsificación de una dirección IP se registra por medio del servicio syslog
Nospoof	Si se utiliza resolución inversa para comparar un nombre de sistema con una dirección indicada, se resuelve el nombre de sistema que se devuelve para verificar que concuerda con la dirección que se consulto. Evita la "falsificación' de las direcciones IP Se activa al indicar nospoof on.
Trim	Toma un nombre de dominio como argumento. trim quita el nombre de dominio antes de ejecutar una consulta /etc/host del nombre. Esto le permite poner solo el nombre base del sistema en /etc/hosts sin especificar el nombre de dominio.
Multi	Toma off u on como argumentos. Se usa para determinar si un sistema esta autorizado a tener mas de una dirección IP indicada en /etc/hosts. multi solo se usa juntamente con consultas nodo. Esta opción no tiene efecto sobre las consultas NIS o DNS.

Tabla 3 12 Opciones de configuración del archivo /etc/host.conf

tristar.com El agente de resolución consulta los nombres de sistema usando primero DNS y

después prueba en el archivo /etc/host local.

La posibilidad de múltiples direcciones IP para un solo sistema esta desactivada. Este sistema comprueba la falsificación de direcciones IP volviendo a resolver el nombre del sistema que devuelve una consulta de dirección IP inversa. Esto se podría considerar como un sobreesfuerzo, pero ayuda a garantizar que nadie esta pretendiendo ser un sistema diferente del que realmente es. Además, se ha configurado el agente de resolución para que le avise de un intento de engaño. Finalmente el agente de resolución depura el dominio triscar.com de cualquier nombre de sistema que fuese buscado en el archivo /etc/host local.

3.3.3.2 EL ARCHIVO */etc/resolu.conf*

Una vez configurado el comportamiento básico de la biblioteca del agente de resolución, se debe instalar alguna información para la parte DNS del mismo. Solo es necesario hacerlo si se usa DNS para la resolución de nombres de sistema, es decir, al indicar bind en la sentencia *order* del archivo */etc/host conf.*

El archivo */etc/resolv.conf* controla la forma en la que el agente de resolución usa DNS para resolver nombres de sistema. Indica los servidores de nombre DNS a contactar cuando se resuelve un nombre de sistema y en que orden ha de contactarlos. También proporciona el nombre de dominio local y algunas pistas sobre como adivinar el nombre de dominio de sistemas que se indican sin el.

La Tabla 3.13 da una lista de las opciones validas para el archivo */etc/resolv.conf.*

A continuación se muestra un ejemplo del archivo*/etc/resolv.conf* para tristar.com:

Opción	Descripción
Domain	El nombre del dominio local de este sistema. Si no se da, el agente de resolución intenta obtenerlo con la llamada de sistema getdomainname( ).
Nameserver	Indica la dirección IP de un servidor de nombres DNS a contactar para solucionar el nombre. El usuario puede listar hasta tres servidores de nombres usando repetidas veces la opción nameserver. Los servidores de nombres se prueban en el orden listado Debería colocar en primer lugar a su servidor de nombres mas fiable, de forma que las consultas no se pasen de tiempo en un servidor que probablemente este desconectado.
Search	Da una lista de dominios a probar si no se indica ninguno como parte de un nombre de sistema a consultar Si no se da ninguna opción de búsqueda, se crea la lista de dominios usando el dominio local junto con cada dominio superior suyo.

Tabla 3.13 Opciones de configuración para el archivo /etc/resolv conf

 # /etc/resolv.conf for tristar.com

#

#Set our local domain name

domain tristar.com

 # Specify our primary name server

nameserver 166.82.1.3

En este ejemplo el usuario indica el dominio local por medio de la opción domain y lista un servidor de nombres para resolver los nombres de sistema.

Los servidores DNS pueden dejar de funcionar y lo hacen de forma inesperada. Si se confía solo en un servidor DNS para la resolución de nombres, se puede encontrar incapaz de trabajar si este falla.

3.3.4 EL PROCESO DAEMON NAMED

El servidor de nombres DNS bajo Linux lo proporciona el proceso *daemon named.* Este proceso se arranca normalmente durante el arranque y lee su información de configuración en un conjunto de archivos de configuración. Normalmente, *named* se ejecuta hasta que se desconecta el sistema. Una vez que ha arrancado *named* y se ha inicializado con su información de configuración, escribe su ID de proceso en el archivo ASCII */etc/named.pid.* Entonces comienza a escuchar a la espera de peticiones DNS en el puerto de red predeterminado indicado en */etc/ services.*

3.3.4.1 EL ARCHIVO *named.boot*

Normalmente, el primer archivo que lee *named* cuando arranca es */etc/named.boot* . Es un archivo muy pequeño, pero es la clave de todos los otros archivos de configuración usados por *named.* Contiene punteros a los diversos archivos de configuración y a otros servidores de nombres. En el archivo *named.boot,* los comentarios comienzan con un punto y coma y continúan hasta el fin de la línea. Existen varias opciones que se pueden listar en el archivo *named.boot* . La Tabla 3.14 da una lista de estas opciones.

Opción	Descripción
Directory	Es el directorio donde se encuentran los archivos de zona DNS. Puede indicar diferentes directorios usando repetidamente la opción directory Puede dar nombres de rutas de acceso de los archivos como relativos a estos directorios
Primary	Toma un nombre de dominio y uno de archivo como argumentos. La opción primary declara que named esta autorizado en el dominio indicado y provoca que named cargue la información de zona del archivo indicado.
Secondary	Esta opción informa a named para que actúe como servidor secundario en el dominio indicado. Toma como argumentos un nombre de dominio, una lista de direcciones y un nombre de archivo named intenta transferir la información de zona de los sistemas indicados en la lista de direcciones y almacena esta información en el archivo indicado en la línea de opciones. Si named no es capaz de contactar con ninguno de los sistemas, intenta recuperar la información del archivo de zona secundario
Cache	Instala información de antememoria de named. Toma como argumentos un nombre de dominio y uno de archivo. El nombre de dominio se indica normalmente como .. El archivo contiene un conjunto de registros, conocidos como "pistas" del servidor, que listan información sobre los servidores de nombres raíz.
Forwarders	Toma una lista de servidores de nombres como argumentos Informa al servidor de nombres local para que intente contactar con los servidores en esta lista si no es capaz de solucionar una dirección a partir de su información local.
Slave	Convierte al servidor de nombres local en un servidor esclavo. Si se da la opción slave, el servidor local intenta solucionar los nombres DNS por medio de consultas recursivas Simplemente remite la petición a uno de los servidores listados en la línea de opciones forwarders.

Tabla 3 14 Opciones de configuracion del archivo named.boot

El siguiente es un ejemplo de archivo named.boot:

;named.boot file

; A sample named.boot for tristar.com

;

directory /var/named

;

cache.named.ca

primary tristar.com.named.hosts

primary 197.198.199 in-addr.arpa named.rev

En este ejemplo se instala el servidor de nombres primario de tristar.com. Como se puede observar, los comentarios comienzan con el carácter ";". La sentencia *directory* en el archivo indica a *named* que todos los archivos de trabajo se encuentran en el directorio */var/named.* Debido a que ninguno de los otros archivos nombrados en el archivo *named.boot* tiene rutas de acceso de directorio asociadas con ellos, se encuentran en */var/named.*

La línea siguiente instala la información de antememoria de este servidor de nombres. Esta opción debería estar presente en casi todos los sistemas que se ejecuten como un servidor de nombres, informa a *named* para que active la antememoria y cargue la información del servidor raíz desde el archivo named.ca.

La entrada cache es muy importante, sin ella no se activa la antememoria del servidor de nombres local. Esto puede producir graves problemas de rendimiento en las búsquedas de nombres. Además, el servidor local no puede contactar con ningún servidor de nombres raiz y, a consecuencia de esto, no es capaz de solucionar ningún nombre de sistema no local, a menos que este instalado como un servidor de nombres que remita a otros.

La siguiente línea en el archivo *named.boot* informa a *named* de que este servidor tiene autoridad primaria en el dominio tristar.com. Los registros de zona y de información del sistema están en el archivo *named.hosts* .

Existe una segunda línea primary en el archivo *named.hosts*, esta línea le muestra que también tiene autoridad de zona primaria en la zona 197.198.199.in-addr.arpa con información de zona en el archivo *named.rev*. Esta extraña sintaxis es la forma en la que *named* obtiene información para comparar las direcciones IP con los nombres DNS. Debido a que DNS fue originalmente instalado para comparar nombres DNS con direcciones IP, se necesita una línea primary diferente para realizar la resolución inversa.

3.3.4.2 ARCHIVOS DE LA BASE DE DATOS Y REGISTROS DE RECURSOS

Toda la información en los diversos archivos de la base de datos de *named* se almacena en un formato conocido como registro de recursos. Cada registro de recursos tiene un tipo asociado con el, que informa de la función del registro. Un registro de recursos es la menor cantidad de información que usa *named*.

La mayoría de la gente encuentra que la sintaxis de los registros de recursos y de los archivos maestros de base de datos en general es poco clara. No ayuda nada el hecho de que algunos registros de recursos han de aparecer en ciertos lugares de ciertos archivos.

Dentro de los archivos maestros de configuración, se tiene la opción de indicar los nombres de sistemas absolutos o los nombres de sistemas que están relacionados con este dominio. Los nombres de sistemas se consideran absolutos si acaban con un carácter ".". Los nombres de sistemas que no terminan con un carácter punto se consideran relacionados con el dominio local, conocido también como origen. Puede referirse al mismo origen usando el carácter @.

Los registros de recursos usan una sintaxis general que consiste en todos los tipos de estos registros. Sin embargo, para aumentar mas la confusión hay varias partes del registro que son opcionales dependiendo del tipo de registro y pueden tomar un valor predeterminado si no se indican. El formato básico del registro de recursos es

.ifconfig lo 127.0.0.1

Los campos están separados por espacios en blanco: espacios o tabuladores. La Tabla 3.15 describe que significan los diversos campos.

Campo	Descripción
Owner	El nombre de dominio o de sistema al cual se aplica el registro. Si no se da ningún nombre se supone el nombre de dominio del registro de recursos previo
ttl	El campo de tiempo de expiración. Este campo informa sobre cuánto tiempo, en segundos, es valida la información del registro después de haberse recuperado desde el servidor DNS. Si no se da ningún valor a ttl, se usa el ttl mínimo del último registro de Comienzo de Autoridad (SOA).
Class	Indica una clase de dirección de conexión en red Para las redes TCP/IP use el valor IN Si no se da la clase, se usa la del registro de recursos previo
Type	Lista el tipo de registro de recursos. Este valor es obligatorio. Los diversos tipos de registros de recursos se listan en la siguiente sección.
Data	Indica los datos asociados con el registro de recursos Este valor es obligatorio. El formato del campo data depende del contenido del campo type.

Tabla 3.15 Campos en el formato de datos del registro de recursos.

CAPITULO 4

ADMINISTRACION DE LOS SISTEMAS DE ARCHIVOS

4 ADMINISTRACIÓN DE LOS SISTEMAS DE ARCHIVO

4.1 ADMINISTRACIÓN DE LOS SISTEMAS DE ARCHIVO

Los sistemas de archivo son la base de todos los datos en Linux. Todos los programas de Linux. las bibliotecas, los archivos del sistema, y los de usuario residen en los sistemas de archivos. Por lo que la adecuada administración de estos es critica porque todos los datos y programas están en esos sistemas de archivos.

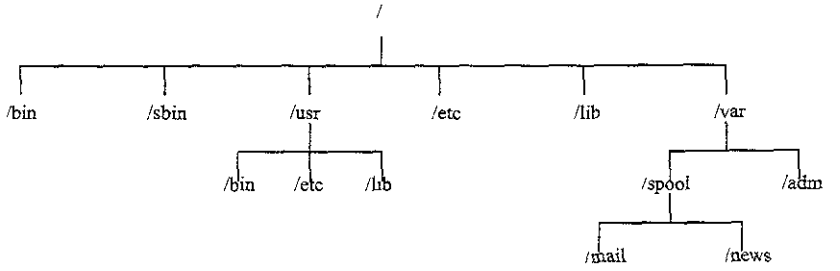

Fig 4.1 Estructura de árbol Linux.

4.1.1 QUE ES UN SISTEMA DE ARCHIVOS

Bajo Linux, el espacio de archivo que resulta visible para los usuarios se basa en una estructura en árbol, con la raíz arriba del todo. Los distintos directorios y archivos se ramifican hacia abajo desde la raíz. El directorio superior (/) se conoce como el directorio raíz. La figura 4.1 muestra un ejemplo de una estructura en árbol.

Desde el punto de vista del usuario este árbol parece una entidad uniforme; lo único que se ven son directorios y archivos. En realidad muchos de los directorios que se ven en el árbol están situados como distintas particiones de disco, en diferentes discos e incluso en distintas computadoras. Cuando una de esas particiones de disco se anexa al árbol en un directorio

conocido como punto de montaje, este y todos los directorios por debajo de el se denominan "sistema de archivos"

4.1.2 EL SISTEMA DE ARCHIVOS LINUX

El sistema operativo Linux se compone de varios directorios y de muchos archivos distintos. Normalmente la mayor parte de el sistema operativo reside en dos sistemas de archivos; el sistema de archivos raíz, conocido como (/) y el sistema de archivos montado bajo */usr*, llamado user (usuario).

Entre los directorios de Linux, podemos mencionar los siguientes:

El directorio */bin* tiene programas ejecutables, conocidos como binarios. Estos son programas esenciales para el sistema y muchos de los comandos son en realidad programas que están en este directorio.

El directorio */sbin* también se utiliza para almacenar archivos binarios del sistema. La mayor parte de los archivos de este directorio se utilizan para administrar el sistema.

El directorio */etc* es muy importante y contiene muchos de los archivos de configuración del sistema. Esencialmente, estos son los archivos que forman la personalidad de Linux. Aquí también se encuentra el archivo de contraseñas. Además, este directorio contiene los archivos de ordenes de inicio para Linux, la lista de sistemas principales con direcciones IP que se quieren registrar permanentemente y muchos otros tipos de información de configuración. Las bibliotecas compartidas que utilizan los programas al ejecutarse se almacenan en el directorio */lib*. Por medio del uso de bibliotecas compartidas muchos programas pueden reutilizar el mismo código y, como esas bibliotecas pueden ser almacenadas en un lugar común, queda reducido al tamaño de los programas en tiempo de ejecución.

El directorio */dev* contiene archivos especiales que se conocen como archivos de dispositivos. Estos se utilizan para acceder a todos los distintos tipos de hardware que hay en el sistema. Por ejemplo, el archivo */dev/mouse* sirve para leer las entradas del ratón. Al organizar a los dispositivos hardware de este modo, Linux hace que la interfaz correspondiente parezca un archivo. Lo que esto significa es que, en muchos casos se puede utilizar la misma sintaxis usada en los archivos, para realizar operaciones en los dispositivos hardware de las computadoras.

Muchos de los dispositivos en el directorio */dev* están en grupos lógicos. La tabla 4.1 muestra algunos de los dispositivos utilizados mas frecuentemente del directorio */dev*.

Subdirectorio	Descripción
/dev/console	Hace referencia a la consola del sistema, que es el monitor de la computadora conectada físicamente al sistema Linux.
/dev/hd	La interfaz de dispositivo para las unidades de disco duro I(DE. El dispositivo /dec/hda1 hace referencia a la primera partición de disco duro hda. El dispositivo /dev/hda hace referencia al disco duro hda en su totalidad.
/dev/sd	Esta es la interfaz de dispositivo para discos SCSI. En los discos y particiones se aplican los mismos convenios que para los dispositivos /dev/hd
/dev/fd	Estos dispositivos proporcionan soporte a unidades de disquete /dev/fd0 es la primera unidad de disquete y dev/fd1 la segunda.
/dev/st	Este es el dispositivo para unidades de cinta SCSI
/dev/sr	Este dispositivo proporciona la interfaz para unidades CD-ROM SCSI
/dev/tty	Estos dispositivos proporcionan distintas consolas para las entradas de usuario. El nombre proviene de cuando la terminales, llamadas teletipos, estaban físicamente enlazadas a un sistema Unix Bajo Linux proporcionan soporte para las consolas virtuales. Estas consolas virtuales proporcionan sesiones independientes de entrada local simultánea.
/dev/pty	Los dispositivos pty proporcionan soporte para pseudo terminales Estas se utilizan para sesiones de entrada remotas, como por ejemplo las que utiliza telnet.
/dev/ttys	Los dispositivos ttys son los puertos interfaz serie de la computadora
/dev/cua	Estos son unos dispositivos especiales de llamada que se utilizan con modems.

Tabla 4.1 Dispositivos demayor uso del directorio /dev

El directorio */proc* es realmente un sistema de archivos virtual. Se utiliza para leer información de los procesos desde memoria.

El directorio */tmp* se utiliza para almacenar archivos temporales que crean los programas al ejecutarse.

El directorio */home* es el directorio base para los directorios iniciales de los usuarios.

El directorio */var* contiene archivos que tienen tendencia a cambiar de tamaño con el tiempo. Normalmente varios registros del sistema están ubicados bajo este directorio.

El directorio */usr* y sus subdirectorios son muy importantes para el funcionamiento del sistema Linux. Este directorio contiene varios subdirectorios, que a su vez contienen los programas mas importantes en el sistema. Normalmente, los subdirectorios de */usr* contienen los paquetes de software que se instalan. La tabla 4.2 describe algunos de los subdirectorios de */usr*. El directorio */usr* siempre se monta como un sistema de archivos separado.

4.1.3 EL SISTEMA DE ARCHIVOS DE RED

El sistema de archivos de red (NFS) es un sistema que permite montar sistemas de archivos desde una computadora distinta sobre una red TCP/IP. Bajo NFS se monta localmente un sistema de archivos de una computadora remota y de cara a los usuarios parece igual que un sistema de archivos local. Esto tiene muchas aplicaciones, por ejemplo, se puede tener una maquina en la red con mucho espacio en disco y que actúa como servidor. Esta computadora tiene todos los directorios de inicio de todos los usuarios en sus discos locales. Si esos discos se montan vía NFS en todas las demás computadoras, los usuarios pueden acceder a sus directorios iniciales desde cualquier computadora en la red.

Subdirectorio	Descripción
/usr/bin	Este subdirectorio se utiliza para tener muchos de los programas ejecutables del sistema Linux instalados.
/usr/etc	Este directorio contiene muchos archivos misceláneos de configuración del sistema
/usr/include	Aquí y en sus subdirectorio es en donde están todos los archivos include para el compilador C. Estos son archivos de cabecera que definen constantes y funciones que son criticas para la programación en C.
/usr/g++-include	Contiene los archivos include para el compilador C++
/usr/lib	Contiene varias bibliotecas para que las utilicen los programas durante el enlace.
/usr/man	Contiene las diversas paginas del comando man para los programa en el sistema Linux. Debajo de /usr/man hay varios directorio que se corresponden con las distintas secciones de las paginas del comando man.
/usr/src	Este directorio contiene directorios que tienen código fuente de distintos programas en el sistema.
/usr/local	Este directorio esta asignado para personalizaciones locales del sistema Generalmente la mayor parte del software local se instala en los subdirectorios de /usr/local.

Tabla 4 2 Subdirectorios importantes en el sistema de archivos /usr

Hay tres componentes esenciales en NFS. En primer lugar las computadoras que contienen los sistemas de archivos que se quieran montar con NFS tienen que poder comunicarse unos con otros por medio de una red TCP/IP. En segundo lugar, la computadora que tiene el sistema de archivos que se pretende sea local debe tener ese sistema de archivos disponible para ser montado. A esta computadora se le conoce como servidor y al proceso de hacer disponible el sistema de archivos se le conoce como exportación. En tercer lugar, la computadora que quiere montar el sistema de archivos exportado, conocido como cliente, tiene que montarlo como un sistema de archivos NFS.

4.2 COMPRENSIÓN DEL SISTEMA DE ARCHIVOS Y DIRECTORIOS

4.2.1 COMPRENSIÓN DE LOS NOMBRES DE ARCHIVO

En Linux, al igual que en cualquier otro sistema operativo, como MS-DOS, es necesario distinguir entre un nombre de archivo y un nombre de ruta de acceso. Un nombre de archivo consiste en una serie de letras, números y ciertos signos de puntuación. Los nombres de archivo no pueden tener espacios o cualquier carácter que represente un separador de campo. Por ejemplo el nombre de archivo 'ejemplo.txt' es válido, pero el nombre 'ejemplo txt' no lo es.

Los nombres de los archivos no deberán contener ningún carácter que tenga un significado especial para el shell. Esos caracteres especiales son los siguientes:

! @ # $ % ^ & * ( ) [ ] { } ' " \ / | ; < >

Los nombres de los archivos tampoco pueden tener el carácter (/) porque éste se utiliza para indicar nombres de ruta de acceso.

En realidad, se puede utilizar cualquiera de esos caracteres marcando entre comillas el nombre del archivo, por ejemplo "! tesis.txt", pero va a ser difícil acceder el archivo con la mayor parte de los programas, y el archivo no es muy portable a otros sistemas UNIX.

La mayor parte de las primeras versiones de UNIX, en el que se basa Linux, limitaban la longitud de los nombres de archivo a 14 caracteres; sin embargo Linux permite hasta 256 caracteres en esos nombre. Algunas versiones recientes de UNIX, como, por ejemplo, la

versión Berkeley de UNIX (BSD) permiten nombres de 64 caracteres, pero sólo son significativos los primeros 14.

Un *nombre de ruta de acceso* puede ser cualquier número de caracteres. En Linux un archivo no existe en un espacio vacío, sino que está dentro de un directorio. Al directorio superior se le *llama directorio raíz* y se simboliza por el carácter de barra inclinada (/); se le conoce simplemente como raíz. Si, por ejemplo, hay un archivo llamado ejemplo en el directorio raíz, el nombre completo de ruta de acceso es */usuario1*. Cuando se añade un usuario al sistema también se le asigna un directorio inicial. Por convenio, este directorio inicial se encuentra normalmente bajo el raíz, en un directorio llamado *home*. Si a un usuario se le asigna un directorio llamado */home/usuario1*, todos los archivos que crea ese usuario se anexan a ese directorio. El nombre de ruta de acceso absoluta para uno de los archivos de el usuario podría ser */home/usuario1/usuario1.file*. Un nombre de ruta de acceso absoluta especifica exactamente donde se puede encontrar el archivo en el sistema de archivos.

Hay otra clase de nombre de ruta de accèso: un nombre de ruta de acceso relativa. Un nombre de acceso relativa apunta sin ambigüedad a un archivo relativo al directorio actual. Si el *usuario1* está en su directorio de usuario, el nombre de archivo *usuario1*.file es también un nombre de ruta de acceso relativa, respecto a su directorio actual.

Se puede definir un archivo con nombres de ruta de acceso relativos utilizando los dos seudónimos que se encuentran en todos los directorios: el punto (.) se refiere al directorio actual y el punto doble (..) se refiere al directorio superior. MS-OS y OS/2 utilizan los mismos convenios.

Si el usuario está en "home/usuario1" puede apuntar a */usuario1* mediante la utilización de *../../usuario1*. El primer conjunto del punto doble señala a */home* (el directorio padre de */home/usuario1*) y el segundo señala al directorio superior de */home*, que es el raíz. El seudónimo del directorio actual, el punto solo, es muy útil si se quiere mover archivos. Si el

usuario quiere mover */usuario1* a su directorio actual, puede hacerlo con nombres de ruta de acceso absolutos mediante la utilización del comando que se indica a continuación:

mv /usuario1 usuario1

De forma alternativa, el usuario1 puede utilizar el seudónimo del directorio actual utilizando este comando:

mv /usuario1 .

La mayoría de los comandos de Linux trabajan con nombres de rutas de acceso. En la mayor parte de los casos, el nombre de la ruta de acceso que se utiliza es el nombre de un archivo en el directorio actual. El nombre de ruta de acceso predeterminado señala al directorio actual. Si el usuario está en su directorio inicial, */home/usuario1*, los tres ejemplos siguientes son equivalentes:

command usuario1 letter
command /home/usuario1/usuario1.letter
command /usuario.letter

Aunque no son lo mismo los nombres de archivos que los nombres de rutas de acceso, también los directorios son archivos después de todo. Cuando se ponga nombre a los directorios, se debe recordar que tienen las mismas limitaciones que los archivos normales.

También debe notarse que, a diferencia de muchos sistemas operativos basados en PC, Linux no tiene el concepto de letras de unidades de disco, sino sólo de rutas de acceso al directorio.

4.2.1.1 REVISIÓN DE LOS TIPOS DE ARCHIVO

Hay cuatro tipos básicos de archivos: archivos normales, directorios, enlaces y archivos especiales. Hay varias clases de archivos normales, enlaces y archivos especiales y un gran número de directorios estándar. Cada uno de estos se describe en las secciones siguientes.

4.2.1.1.1 ARCHIVOS NORMALES

Los archivos normales son con los que se trabaja la mayor parte del tiempo. Los archivos normales pueden contener texto, código fuente en lenguaje C, archivos de órdenes shell (programas interpretados por uno de los shell de Linux), programas binarios ejecutables y datos de diversos tipos. Para Linux, un archivo no es nada más que un archivo. La única diferencia es que Linux sabe cuales de sus archivos están marcados como ejecutables. Los archivos ejecutables pueden ser ejecutados directamente, siempre que contengan algo para ejecutar y que estén en la ruta de acceso de búsqueda. Básicamente la ruta de acceso de búsqueda es una lista de nombres de rutas de acceso que se han especificado en las que Linux busca para encontrar un archivo ejecutable.

Los archivos ejecutables son archivos binarios, es decir archivos que ejecutan código máquina y archivos de órdenes shell. El comando *file* de Linux revisa los datos de un archivo y averigua lo que hay dentro. Si se escribe *file* *, se puede obtener algo similar a lo que se muestra a continuación:

INSTALL: *symbolic link to /var/adm*
ghostvw.txt: *ascii.text*
Linux *symbolic link to /usr/src/linux*
mbox: *mail text*
mterm.txt *English text*
seyon.txt: *English text*
xcalc.txt *English text*

xclock.txt. *English text*
xeyes txt· *English text*
xgrap txt· *English text*
xlock txt· *English text*
xspread.txt: *English text*
xtris.txt· *empty*

Todos los archivos de la primera columna son archivos ordinarios que contienen distintos tipos de datos y todos ellos están ubicados en un directorio.

4.2.1.1.2 ARCHIVOS DE DIRECTORIO

Los directorios son archivos que contienen los nombres de archivos y subdirectorios, así como punteros hacia esos archivos y subdirectorios. Los archivos de directorio son el único sitio donde Linux almacena nombres de archivos. Cuando se lista el contenido de un directorio con el comando *ls,* lo único que se hace en realidad es listar el contenido del archivo de directorio.

Cuando se cambia el nombre de un archivo con el comando *mv* y este archivo está en el directorio actual, todo lo que se está haciendo es cambiar la entrada en el archivo de directorio. Si se mueve un archivo desde un directorio a otro, lo único que en realidad se está haciendo es mover la descripción del archivo, siempre que, naturalmente, el nuevo directorio esté en la misma partición o en el mismo disco físico.

4.2.1.1.3 DIRECTORIOS Y DISCOS FÍSICOS

A cada archivo se le asigna en Linux un número único llamado inode. El inode se almacena en una tabla que se llama la tabla de inodes, que se asigna cuando el disco está formateado. Cada disco físico o partición tiene su propia tabla de inodes. Un inode contiene toda la información sobre un archivo, incluyendo la dirección de los datos en el disco y el tipo de archivo. Los tipos de archivo incluyen archivos normales, directorios y archivos especiales.

El sistema de archivos de Linux asigna el número de inode 1 al directorio raíz. Esto da a Linux la dirección en disco del archivo del directorio raíz; este contiene una lista de nombres de archivo, y directorios y sus números inode correspondientes. Linux puede encontrar cualquier archivo en el sistema por medio de la consulta de una cadena de directorios, comenzando por el directorio raíz, cuyo contenido de archivo puede parecerse a lo que se muestra a continuación:

1 .
2 ..
45 *etc*
230 *dev*
420 *home*
123 *.profile*

Debe observarse que los archivos (.) (punto) y (..) (punto doble) están representados en el directorio. Debido a que éste es el directorio raíz, (.) , y su directorio padre, (..), son idénticos. El contenido del archivo de directorio */home* es distinto.

420 .
1 ..
643 *usuario1*

Obsérvese que el inode del directorio actual (.) coincide con el inode para */home*, que se encuentra en el archivo del directorio raíz, y que el inode para el directorio padre (..) es el mismo que el del directorio raíz.

Linux navega por su sistema de archivos por medio del encadenamiento hacia arriba y hacia abajo del sistema de archivos de directorio. Si se quiere mover un archivo a un directorio en

otro disco físico, Linux detecta esto al leer la tabla de inodes. En este caso, el archivo se mueve al nuevo disco, donde se le asigna un nuevo inode antes de suprimirlo de donde estaba originalmente.

De igual manera que con el comando mv, cuando se suprime un archivo con el comando rm en realidad nunca se toca el archivo, sino que Linux marca ese inode como libre y lo devuelve al conjunto de inodes disponibles. Luego se borra la entrada del archivo en el directorio.

4.2.1.1.4 ENLACES

Los enlaces normales no son archivos en absoluto, sino simplemente entradas de directorio que señalan al mismo inode. La tabla de inodes sigue la pista de todos los enlaces que hay con un archivo y sólo cuando se suprime la última entrada de directorio se deja libre el inode para situarlo en el conjunto de inodes disponibles. Naturalmente los enlaces ordinarios no pueden realizarse más allá de los límites del dispositivo porque todas las referencias de directorio señalan el mismo inode.

Linux, al igual que la mayor parte de las versiones modernas de UNIX, tiene otra clase de enlace llamado enlace simbólico, para el cual la entrada de directorio contiene la entrada de un archivo que en sí mismo es una referencia a otro archivo que está ubicado en otro sitio en el sistema de archivos lógico de Linux. Un enlace simbólico puede señalar a otro archivo o directorio en el mismo disco, en otro disco o a un archivo o directorio en otra computadora. Una diferencia importante entre el enlace normal y el simbólico es que con los enlaces normales cada uno tiene la misma categoría (es decir, el sistema trata cada enlace como si fuera el archivo original) y no se suprimen los datos en sí hasta que no se suprime el ultimo enlace de ese archivo. Con los enlaces simbólicos, cuando se suprime el archivo original

también se suprimen todos los enlaces simbólicos a ese archivo y, además, estos enlaces no tienen la misma categoría que el original.

Aparte de esas diferencias entre enlaces y archivos, los enlaces se tratan y acceden de la misma forma que si se estuviera accediendo directamente al archivo.

4.2.1.1.5 ARCHIVOS ESPECIALES

En el sistema de archivos se representan todos los dispositivos físicos asociados con Linux, incluyendo discos, terminales e impresoras. La mayoría de los dispositivos están ubicados en el directorio */dev*. Por ejemplo si se está trabajando con la consola del sistema, el nombre asociado de dispositivo es */dev/console*. Si se está trabajando en una terminal estándar, el nombre del dispositivo puede ser */dev/tty0l*. Las terminales, o líneas serie, se llaman dispositivos *tty* (que es la abreviatura de teletipo - teletype -, la terminal original de UNIX).
Las terminales e impresoras se denominan dispositivos especiales por caracteres. Éstos pueden aceptar y producir una cadena de caracteres. Por otro lado los discos almacenan datos en bloques que están direccionados por cilindro y sector. En un disco no se puede acceder sólo a un carácter, sino que hay que leer y escribir bloques completos. Esto mismo sucede normalmente en las cintas magnéticas. A este tipo de dispositivo se le denomina dispositivo especial por bloques. Para poner un poco más de complejidad en todo esto, los discos y otros dispositivos especiales por bloques tienen que ser capaces de actuar como dispositivos orientados a caracteres, de modo que cada dispositivo de bloques tiene su correspondiente dispositivo especial por caracteres. Linux hace la traducción al leer los datos que se envían a un dispositivo por caracteres y traducirlos para un dispositivo por bloques. Esto se hace automáticamente, sin intervención del usuario.

Por lo menos hay otro tipo de dispositivo con el que nos podemos encontrar: un FIFO (memoria intermedia first-in-first-out - "primero en entrar primero en salir") que también se conoce como conducción con nombre. Los FIFOs parecen archivos normales; si se escribe en

ellos crecen, pero cuando se leen encogen. Se utilizan principalmente en procesos del sistema para que muchos programas puedan enviar información a un solo proceso de control. Por ejemplo, cuando se imprime un archivo con el comando *lp*, éste establece el proceso de impresión y señala el proceso 'daemon' lpsched por medio del envío de un mensaje a FIFO. Un proceso 'daemon', a veces llamado demonio, es un proceso del sistema que actúa sin necesidad de petición por parte del usuario.

Hay un archivo de dispositivo especial muy útil: El cubo de bits, */dev/null*. Cualquier cosa que se envíe a */dev/null* es ignorada, cosa que resulta muy útil cuando no se quiere ver la salida de un comando. Por ejemplo, si no se quieren ver los informes de diagnóstico impresos en el dispositivo estándar de error, se pueden poner en el cubo de bits utilizando el comando siguiente:

ls -la> /dev/null

4.2.1.1.6 PERMISOS DE LOS ARCHIVOS

Los permisos de los archivos significan más en Linux que simplemente saber los permisos que se tienen en un archivo o directorio. Aunque los permisos deciden quien puede leer, escribir o ejecutar un archivo, también deciden el tipo de archivo y cómo se ejecuta el archivo.

Pueden mostrarse los permisos de un archivo con el comando *ls -l*. Con este comando se podrá ver un listado de directorio similar al que se muestra a continuación:

drw---	*2*	*sglines*	*doc*	*512*	*Jan*	*13.34*	*Mail*
drwx---	*5*	*sglines*	*doc*	*1024*	*Jan*	*08:22*	*News*
-rw---	*1*	*sglines*	*doc*	*1268*	*Dec*	*15·01*	*biblio*
drw---	*2*	*sglines*	*doc*	*512*	*Dec*	*21.28*	*bin*

-rw---	*1*	*sglines*	*doc*	*44787*	*Oct*	*06:59*	*books*
-rw---	*1*	*sglines*	*doc*	*23001*	*Dec*	*2 50*	*bots.msg*
-rw-r---	*1*	*sglines*	*doc*	*105990*	*Dec*	*21:24*	*ducke.gif*

Este listado muestra prácticamente todo lo que se puede saber acerca del archivo desde la entrada del directorio y el inode correspondientes. La primera columna muestra los permisos del archivo, la segunda muestra el número de enlaces a un archivo (o bloques extra en el directorio) y la tercera muestra el propietario del archivo. (En Linux el concepto de propiedad tiene tres posibilidades: el propietario, el grupo del propietario y todos los demás). La cuarta columna muestra el grupo al que pertenece el archivo. La quinta muestra el número de bytes en el archivo y la sexta muestra la fecha y hora de creación: por último, la séptima muestra el nombre del archivo.

La columna de permisos (la primera) se divide en cuatro subcampos:

- rwx rwx rwx

El primer subcampo define el tipo de archivo. Un archivo normal tiene un guión (-) como espacio de reserva; los directorios se marcan con a, b, c y d. La Tabla 4.3 muestra los valores permitidos para el subcampo de tipo de archivo.

Carácter	Significado
-	Archivo normal
b	Archivo especial por bloques
c	Archivo especial por caracteres
d	Directorio
l	Enlace simbolico

Tabla 4 3 Entradas validas en el subcampo tipo de archivo.

Los tres subcampos siguientes muestran los permisos de lectura, escritura y ejecución del archivo. Por ejemplo, *rwx* en el primero de esos subcampos significa que el archivo tiene

permisos para el propietario de lectura, escritura y ejecución. El siguiente subcampo muestra la misma información para la propiedad de grupo del archivo; el tercer subcampo muestra los permisos permitidos para todos los demás.

Esos campos de permisos pueden mostrar más información; de hecho, hay varios atributos empaquetados en esos tres campos. Desafortunadamente, el significado de esos atributos depende de la versión de Linux que se utiliza y de si el archivo es o no es ejecutable.

En estos campos también se puede establecer el bit adosado. Este bit indica al sistema que guarde una copia del programa en ejecución después de que termine. Si el programa se ejecuta con frecuencia, el bit adosado puede ahorrar tiempo al sistema porque el programa no tiene que volverse a cargar en memoria desde el disco cada vez que alguien lo ejecuta.

Normalmente un programa que está en ejecución pertenece a quien lo esté ejecutando. Si está activado el bit de identificador de usuario, el programa en ejecución pertenece al propietario del archivo. Esto quiere decir que el programa en ejecución tiene todos los permisos del propietario del archivo. Si el usuario que ejecuta el programa es un usuario normal y el programa es propiedad del usuario root, el programa de forma automática tiene permiso para leer y escribir cualquier archivo en el sistema sin tener en cuenta los permisos del usuario que lo ejecuta. lo mismo pasa con el bit de establecer el identificador de grupo.

Los permisos de un archivo pueden ser de dos formas distintas: absoluta y relativa. Con los permisos absolutos se definen exactamente permisos del archivo en octal. Un número octal puede tener un valor entre 0 y 7. Originalmente UNIX se creó en una serie de minicomputadoras DEC que utilizaban un sistema octal, de aquí proviene el uso actual de números octal. Los permisos que se quieren poner se agregan conjuntamente para formar un número que los define a todos ellos. La Tabla 4.4 lista los permisos válidos en octal.

Valor octal	Permisos otorgados
0001	Permiso de ejecución para el propietario
0002	Permiso de escritura para el propietario
0004	Permiso de, lectura para el propietario
0010	Permiso de ejecución para el grupo
0020	Permiso de escritura para el grupo
0040	Permiso de lectura para el grupo
0100	Permiso de ejecución para todos los demás
0200	Permiso de escritura para todos los demás
0400	Permiso de lectura para todos los demás
1000	Bit adosado activo
2000	Bit de identificador de grupo activo si el archivo es ejecutable; de lo contrario, se activa el bloqueo de archivos obligatorio
4000	Bit de identificador de usuario activo si el archivo es ejecutable

Tabla 4.4 Permisos octal absolutos utilizados con el comando chmod

Los identificadores de grupo y usuario se refieren a quién tiene permiso de utilizar, leer o ejecutar un archivo. Esos permisos iniciales de archivo los otorga el administrador del sistema cuando se crea la cuenta del usuario. Sólo los usuarios de un grupo determinado pueden acceder a los archivos de un grupo y sólo si el usuario ha dado permiso a los miembros del grupo para esos archivos.

Para dar permisos de lectura y escritura para todo el mundo, hay que agrear los permisos conjuntamente, como muestra el ejemplo que se indica a continuación:

0002 Write permission for the owner

0004 Read permission for the owner

0020 Write permission for the group

0040 Read permission for the group

0200 Write permission for all others

0040 Read permission -for all others

0666 Read and write permission for everyone

Para dar esos permisos a un archivo se debe utilizar el comando siguiente:

chmod 666 file

Los permisos relativos utilizan un formato ligeramente distinto. Con este tipo de permisos hay que establecer lo siguiente:

- A quién se están dando permisos
- Que clase de operación se trata de hacer (agregar, sustraer o establecer permisos)
- Cuáles son los permisos

Por ejemplo, si se escribe a *chmod a=rwx* archivo, se está dando permiso de lectura, escritura y ejecución a todos los usuarios. Los comandos se resumen en la tabla 4.5

Valor	Descripción
Quién	
A	Todos los usuarios (el usuario, sus grupos y todos los demás)
G	El grupo del propietario
o	Todos los demás
u	Sólo el usuario
Operador	
+	Agrega la modalidad
-	Elimina la modalidad
=	Establece la modalidad de forma absoluta
Permiso	
x	Establece la ejecución
r	Establece la lectura
w	Establece la escritura
s	Establece el bit de identificador de usuario
t	Establece el bit adosado

Tabla 4.5 Permisos relativos utilizados con el comando chmod

"Si se ha marcado un archivo con el bit de identificador de usuario activo, los permisos que se muestran con el comando *ls -l* aparecen conio se indica a continuación:

-rws- - - - - 1 sglines 3136 jan 17 15:42 x

Si se agrega el bit de identificador de grupo, los permisos aparecen como se indica a continuación:

-rws- - S - - - 1 sglines 3136 jan 17 15:42 x

Si entonces el bit adosado se pone activo, los permisos aparecen como sigue:

-rws- - S - - rws - - S - - T 1 sglines 3136 jan 17 15:42 x

Obsérvese el uso de las mayúsculas S y T para indicar el estado de los diversos bits."[13]

4.2.2 REVISIÓN DE LOS DIRECTORIOS ESTÁNDAR DE LINUX

Hay un conjunto clásico de directorios para UNIX y lo que puede ser llamado "el conjunto estándar de directorios emergentes", que es lo que Linux básicamente sigue.

[13] Tackett & Gunter, Utilizando Linux 2ª edicion, Prentice Hall, PP 362,363.

4.2.2.1 DIRECTORIOS UNIX CLÁSICOS

Antes de UNIX System V versión 4 (por ejemplo, UNIX System V versión 3.2 y anteriores), la mayor parte de las versiones de UNIX se establecían en un sistema regular de organización de los directorios UNIX parecido al que se muestra a continuación:

/

/etc

/lib

/tmp

/bin

/usr

/spool

/bin

/include

/tmp

/adm

/lib

El directorio */etc* contiene la mayor parte de los datos específicos del sistema para arrancar, o para hacer que el sistema esté activo. Contiene archivos tales como passwd e inittab, que se necesitan para un funcionamiento adecuado del sistema.

El directorio */lib* contiene una biblioteca de funciones necesaria para el compilador C. Este directorio es importante aunque no se tenga este compilador en el sistema, porque contiene todas las bibliotecas compartidas a las que pueden llamar los programas de aplicación. Una biblioteca compartida se carga en la memoria sólo cuando se ejecuta el comando que la llama.

Esta disposición permite que los programas ejecutables sean de menor tamaño, porque, de otro modo, cada programa contendría código duplicado, necesitando más espacio en disco para almacenarse y mucha más memoria para ejecutarse.

El directorio */tmp* se utiliza para almacenamiento temporal. Los programas que utilizan */tmp* por lo general hacen limpieza después de ejecutarse y suprimen cualquier archivo temporal. No se debe guardar ninguna cosa que se necesite en este directorio, debido a que el sistema suprime periódicamente su contenido de forma automática.

El directorio */usr* contiene todo lo demás. La variable PATH contiene la cadena */bin: /usr/bin* porque el directorio */usr/bin* contiene todos los comandos Linux que no están en el directorio */bin*. Este tipo de disposición tiene un precedente histórico. Al principio de Linux, los discos duros no tenían mucha capacidad. Linux necesita por lo menos los directorios */etc/tmp/* y */bin* para el procedimiento de rutina de carga (es decir, para comenzar a ejecutarse). Debido a que los discos de la primera época de Linux tenían sólo esos tres directorios, todo lo demás estaba en otro disco que podía montarse sólo después de que Linux estuviera activo y en ejecución. El hecho de colocar directorios adicionales en el directorio */usr* no suponía mucho problema cuando Linux aún era relativamente un sistema operativo pequeño y permitía que un sistema Linux de tamaño moderado existiera con sólo dos discos: Un disco raíz y el disco */usr*.

El directorio */usr/adm* contiene toda la información de registro y diagnostico que necesita el administrador del sistema. Este directorio está vacío cuando los programas de registro y diagnostico están desactivados.

El directorio */include* contiene todo el código fuente utilizado por las sentencias #include en los programas C. Deberá tenerse como mínimo acceso de lectura a este directorio porque contiene todos los fragmentos de código y estructuras que definen el sistema.

El directorio */usr/spool* contiene todos los datos temporales utilizados por el sistema de impresión lp, el proceso daemon cron y el sistema de comunicaciones UUCP. Los archivos enviados a la gestión de colas de la impresora se guardan en el directorio */spool* hasta que se imprimen. También se almacena aquí cualquier programa que esté en espera de ser ejecutado por cron, incluyendo todos los archivos crontab y los trabajos pendientes de at y batch.

El directorio */usr/lib* contiene la parte restante que forma el sistema estándar Linux. En general este directorio representa un caos controlado que está escondido debajo del sistema Linux, que está relativamente bien organizado. Este directorio contiene programas que se invocan por otros programas en */bin* y */usr/bin* así como también archivos de configuración para terminales e impresoras, el sistema de correo, cron y el sistema de comunicaciones UUCP.

El directorio */usr* contiene todos los subdirectorios asignados a los usuarios. El convenio general es: si el identificador de entrada es *usuario1*, el directorio de usuario es */usr/usuario1*.

4.2.2.2 LOS DIRECTORIOS DE LINUX

Uno de los problemas de la estructura clásica de Linux es que el hacer copias de seguridad de los archivos de datos es difícil con un directorio */usr* fragmentado. Generalmente en un sistema se necesitan tres niveles diferentes de copia de seguridad: el sistema básico en sí, cualquier cambio hecho en las tablas que definen el sistema básico para una ubicación específica y los datos de los usuarios.

Sólo puede hacerse una copia de seguridad del sistema básico; sólo deben hacerse copias de seguridad de las tablas de control cuando haya cambios en ellas. Sin embargo, los datos de los usuarios cambian constantemente y deberían hacerse con frecuencia copias de seguridad de los mismos. La estructura típica de directorios de Linux aparece como se indica a continuación,

pero la estructura de cada distinto sistema instalado puede ser un poco distinta, dependiendo de los paquetes instalados:

/

/etc

/passwd (the user database)

/rc (the system initialization script)

/sbin

/bin

/tmp

/var

/lib

/home

/<your user name here> (user accountants)

/install

/usr

/bin

/proc

Los directorios */bin*, */etc* y */tmp* tienen la misma función que en la estructura clásica. Las tablas de definición del sistema se han trasladado al directorio */var*, de modo que en cualquier momento en que se cambie la operación del sistema se pueda hacer copia de seguridad de ese directorio.

La diferencia con UNIX es que todos los programas del sistema se han trasladado al directorio */sbin*. Todos los programas estándar de Linux están en */usr/bin*, que está enlazado con */bin*. Por razones de compatibilidad se mantienen todos los directorios clásicos con enlaces simbólicos. El directorio */usr*, que ya no contiene datos de usuario, ha sido reorganizado con

CAPITULO 5

USO DE INTERNET

5 Uso de Internet

5.1 FUNDAMENTOS DE INTERNET

5.1.1 LA ESTRUCTURA DE LA RED INTERNET

Internet es en realidad una red de redes informáticas que intercambian información entre si. De hecho, la palabra Internet proviene del termino internetwork, que significa "comunicación entre redes". Una forma sencilla de visualizar Internet es pensar en una nube de gran tamaño con computadoras conectadas como se muestra en la figura 5.1 . Esta nube cambia y crece constantemente según se añaden nuevas redes y se modifican las ya existentes.

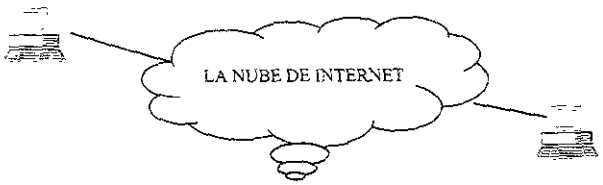

Fig. 5.1 Estructura lógica de la nube Internet.

Dentro de esta nube Internet hay muchas redes conectadas entre si. Estos sistemas utilizan el conjunto de protocolos TCP/IP para comunicación de datos.

5.1.2 HISTORIA

La red Internet surgió de un programa de investigación realizado por la Defense Advanced Research Projects Agency (ARPA) de los Estados Unidos, que se centro en formas de enlazar varias redes informáticas. El resultado de este programa fue ARPANET, lanzada en 1969. En 1971 había aproximadamente 40 computadoras, o sistemas, conectados a ARPANET y los investigadores estaban desarrollando la capacidad de enviar email entre redes. ARPANET continuo creciendo durante los años 70 y otras redes informáticas comenzaron a conectarse también este sistema.

La investigación sobre las comunicaciones entre redes condujo al desarrollo de los protocolos de red TCP/IP, que sustituyeron al anterior conjunto de protocolos llamado NCP, y que se convirtió en el estándar de ARPANET Según mas y mas redes se fueron conectando a ARPANET y entre si, este inmenso complejo de redes paso a ser conocido como Internet. La red original ARPANET se clausuro en 1990, dejando a Internet como su prospero sucesor.

Internet ha experimentado un ritmo increíble de crecimiento. En 1984 había aproximadamente 1,000 sistemas en Internet. En 1989 este numero había crecido a mas de 100,000. Tres años mas tarde, en 1992, el total era de mas de 1 millón de computadoras conectadas a Internet. En julio de 1994 había mas de 3.2 millones de sistemas informáticos conectados a la red, con unos 20 millones de usuarios.

En lo que atañe al tamaño geográfico de Internet, la red en realidad cubre el mundo entero. Casi todos los países industrializados tienen algún tipo de conectividad Internet.

Básicamente existen dos niveles diferentes de conectividad Internet. En el nivel mas interactivo, una ubicación tiene algún tipo de conexión física a una red miembro de Internet. Esta ubicación ejecuta el conjunto de protocolos TCP/IP como sus protocolos de trabajo en red y puede realizar conexiones directas de red con otras computadoras en tiempo real. Esto se denomina estar conectado IP. El otro tipo de conexión normalmente utiliza el protocolo UUCP para transferir correo electrónico y noticias USENET a una procedente de esta. Los usuarios con este tipo de conexión no pueden acceder a otros ordenadores de forma interactiva o en la red.

5.1.3 LOS NOMBRES DE INTERNET

Con millones de usuarios en Internet, ¿como puede especificarse el usuario con el que desea comunicarse? Se Deberá conocer el nombre de la computadora, al igual que el nombre de alguien quien desee enviar un mensaje. Estos nombres son especificados por medio de una convención llamada Sistema de nombres de dominio (DNS) y se encuentran detallados en las Peticiones de Comentarios (RFC) de Internet números 1032, 1033, 1034 y 1035.

Un nombre DNS tiene el formato siguiente:

[subdominio].[subdominio]. [...]. dominio

5.1.3.1 DOMINIOS

Un nombre de dominio normalmente proporciona una estructura jerárquica para una computadora o grupo dentro de una organización. Cada país tiene un dominio de dos letras que se le asigna en base al código definido para su país en el documento 3166 de la Organización Internacional para la Normalización de Estándares (ISO).

Por ejemplo para México, el dominio es MX, Para Estados Unidos es US y para España es ES.

5.1.3.2 SUBDOMINIOS

Los campos de subdominio de un nombre DNS sirven para identificar un ordenador o dirección determinada dentro de un dominio. Los subdominios se hacen mas específicos de derecha a izquierda en el nombre de dominio. El campo de subdominio mas a la derecha, el que se encuentra al lado del campo de dominio, normalmente sirve para indicar una organización determinada dentro de un dominio determinado. El nombre ncsu.edu, por ejemplo, hace referencia la Universidad Estatal de Carolina del Norte, subdominio (ncsu) en el dominio edu. Los campos secundarios de subdominio indican un departamento, grupo o computadora dentro de una organización. Estas agrupaciones de subdominios producen un estructura lógica de árbol dentro del dominio.

Un delimitador especial se utiliza para especificar un usuario en una localización determinada. El símbolo @ se utiliza para especificar un usuario, alias o buzón de correo en el nombre DNS de una ubicación. Por ejemplo, un Usuario tiene la cuenta de usuario "*usuario1*" en la maquina *nux1.somewhere.com*. La dirección completa es:

usuario1@linux1.sales.somewhere.com

El símbolo @ se utiliza para separar el buzón de destino de la computadora de destino.

5.1.4 FUNDAMENTOS BÁSICOS DE LOS NOMBRES INTERNET

Cada computadora tiene solo una dirección IP para cada interfaz física que tenga conectada a una red.

Cuando las computadoras se comunican utilizando TCP/IP utilizan la dirección IP. Los nombres DNS son sencillamente un dispositivo que nos ayuda a recordar los sistemas y las redes a las que están conectados. Originalmente cuando Internet se formo por vez primera, el numero de sistemas en la red era muy pequeño. El resultado era que cada sistema disponía de una lista completa de todos los nombres y direcciones de sistemas en un archivo local. Este sistema se volvió rápidamente inmanejable. Cada vez se añadía un nuevo sistema era necesario actualizar cada archivo de sistema en cada computadora. Con el crecimiento explosivo de Internet los archivos de sistemas llegaron también a ser muy grandes. La asignación de nombres DNS a las direcciones IP se realiza ahora en una base de datos distribuida que utiliza un software especifico para realizar la consulta.

La base de datos distribuida y software que componen el sistema de resolución de nombres de DNS están compuestos de las siguientes partes básicas:

- Espacio de nombre de dominio
- Servidores de nombres
- Resolvedores

Espacio de nombre de dominio. El espacio de nombre de dominio es una especificación de una structura de árbol que identifica un conjunto de sistemas y proporciona información sobre los ismos. Desde el punto de vista conceptual, cada nodo en el árbol cuenta con una base de datos de

información sobre los sistemas bajo su autoridad. Se realizan consultas para intentar obtener la información apropiada de esta base de datos. En términos sencillos, esto es solo el listado de todos los tipos diferentes de información, nombres, dirección IP, alias de correos y restante información disponible para su consulta en el sistema DNS.

Servidores de nombres. Los programas que almacenan los datos localizados en el espacio de nombre de dominio reciben el nombre de servidores de nombres. Cada uno posee información completa sobre un subconjunto del espacio de nombre de dominio e información memorizada sobre otras porciones; además, dispone de información completa sobre su área de autoridad. Esta información autoritaria esta dividida en áreas conocidas como zonas, que pueden dividirse entre varios servidores de nombres a fin de proporcionar servicio redundante para una zona. Cada servidor de nombres posee información sobre otros servidores de nombres responsables de diferentes zonas. Cuando llega una solicitud de información sobre la zona para la que un determinado servidor de nombres es responsable, este sencillamente proporciona la información. Sin embargo, cuando llega una solicitud de información para una zona diferente, el servidor de nombres se pone en contacto con el servidor correspondiente con autoridad sobre dicha zona.

Resolvedores. Los Resolvedores son simplemente programas que extraen información de los servidores de nombres en respuesta a una consulta sobre un sistema en el espacio de nombre de dominio.

Hay muchas formas diferentes de usar todos los aspectos de Internet. Generalmente se pueden usar varios programas diferentes para acceder a estos servicios, cuyas funciones varían mucho.

A continuación se describen algunos de ellos

5.2 SURF EN INTERNET

5.2.1 FTP COMO USUARIO ANÓNIMO

El FTP como usuario anónimo es un sistema que permite conectarse a un programa de transferencia de archivos de muchas computadoras, sin saber ni el nombre de usuario ni la contraseña. Mediante este sistema se puede acceder a archivos y a datos que los administradores del sistema en el sistema remoto hayan puesto a disposición del publico.

5.2.2 ARCHIE

Uno de los mayores problemas del FTP como usuario anónimo es descubrir donde están situados en Internet los archivos de interés. Se podría buscar y buscar y no encontrar nunca lo que se esta buscando. Claramente se necesita algún tipo de sistema para ayudar a que los usuarios encuentren archivos. Fue a partir de esta necesidad que nació el sistema Archie.

Archie es un programa de consulta de bases de datos que contacta con ubicaciones FTP como usuario anónimo en todo el mundo y solicita cada ubicación la lista completa de todos sus archivos. Archie toma esta información y la indexa en su propia base de datos interna. Puede buscar en esta base de datos para localizar archivos en Internet. Debido a que la actualización de la base de datos de Archie es obviamente un proceso que consume tiempo, se actualiza aproximadamente una vez al mes. Esto significa que es posible, aunque muy improbable, que la localización que da Archie no sea correcta.

Se puede contactar con Archie de dos formas distintas: ejecutando un programa de cliente local que se conecte a un servidor de Archie o conectándose directamente con un servidor a través de telnet. En general se debería usar un programa de cliente si puede, porque mantiene mas baja la carga del sistema en los servidores de acceso publico.

Archie es un servicio muy popular. Los diversos servidores de Archie en todo el mundo pueden cargarse mucho. por consiguiente las peticiones pueden tardar un buen rato en completarse. Algunas ubicaciones ponen limites al numero de conexiones simultaneas para evitar que los servidores se vuelvan demasiado lentos. Si se prueba un servidor Archie y se encuentra que esta totalmente cargado, hay intentarlo con otro servidor distinto. Hay varios servidores Archie en todo el mundo que se pueden conectar con los servicios de Archie.

Cada uno de los servidores es ligeramente distinto, pero la mayoría son lo mismo. Una vez conectado obtendrá un indicador como este:

archie>

donde se pueden introducir los comandos de búsqueda. Los distintos servidores tienen diferentes valores de búsqueda predeterminados. Para determinar que instalación por defecto tiene el servidor al que se esta conecta, se usa el comando show search. Este comando devuelve uno de los siguientes valores:

Regex	Archie interpreta su cadena como una expresión regular de UNIX.
Exact	Su cadena de búsqueda debe concordar exactamente con un nombre del archivo.
Sub	Su cadena de búsqueda concuerda si un nombre de archivo lo contiene como subcadena.
Subcase	Similar al tipo de búsqueda sub, excepto que las mayúsculas o minúsculas de la cadena deben concordar.

.2.3 GOPHER

opher fue uno de los primeros servicios de Internet que realizo un intento serio de tener una terfaz amigable con el usuario. Es un servicio de Internet que permite acceder a información alizando selecciones en una serie de menús. Cuando se conecta con una ubicación que ofrece los

servicios de Gopher, se proporciona un menú de las elecciones disponibles. Se Puede realizar la selección desde el menú, sin tener que conocer el nombre o la dirección IP de la ubicación de destino o el directorio y los nombres de los archivos de la información. Gopher lo maneja de forma transparente.

Una desventaja de Gopher es que no existe una lista de temas estándar de los diversos servidores de Gopher. Los administradores de cada servidor Gopher individual han organizado su información a su manera. Esto significa que cada servidor de Gopher al que se acceda tiene distintos temas. Si da la casualidad que el servidor tiene algunos de los mismos temas, hay probabilidades de que se llamen de distintas formas.

No hay fuentes de información en Internet que sean realmente "especificas de Gopher". Cualquier cosa que pueda obtener a través de Gopher se puede conseguir por otros medios como FTP o Telnet. En algunos casos, las ubicaciones pueden haber escogido que los recursos estén disponibles solo a través de Gopher por motivos de seguridad.

Hay dos formas de conectarse al sistema Gopher: instalar un programa de cliente de Gopher local que envía peticiones a un servidor de Gopher remoto o usar telnet para conectarse con un servidor de Gopher de acceso publico remoto.

5.2.4 NOTICIAS USENET

Definido de la forma mas simple, las noticias USENET, netnews o simplemente news, como se le lama comúnmente, es un forum de discusión en línea. Muchas computadoras en el mundo ntercambian trozos de información, llamados artículos, sobre cualquier tema imaginable. Estas omputadoras no están conectadas físicamente a la misma red; están conectados lógicamente para er capaces de intercambiar datos.

os artículos de noticias de USENET se dividen en grupos de noticias por temas. Estos grupos se ividen en jerarquías basadas en distinciones de temas muy generales. La Tabla 5.1 lista las ases de alto nivel principales de grupos de noticias.

Clase	Descripción
alt	Un gran conjunto de temas que no encaja fácilmente en las otras clases (abreviatura de "alternativa")
comp	Muchos temas distintos relacionados con computadoras
misc	Temas varios que no encajan fácilmente en otra categoría
news	Diversos temas relacionados con el mismo sistema de noticias USENET
rec	Temas recreativos y aficiones
soc	Temas sociales
sci	Diversos temas científicos
talk	Temas diseñados para conversaciones en curso

Tabla 5.1 Clases de grupos de alto nivel en las jerarquías USENET

Como todo en Internet, hay excepciones a la reglas previas. Existen muchas mas jerarquías de alto nivel, la mayoría dedicadas a distintas regiones del mundo.

Las noticias USENET plantean conversaciones y discusiones sobre casi cualquier tema que pueda pensar. Es una gran forma de encontrar e intercambiar información.

5.2.5 LISTAS DE CORREO

Las listas de correo e-mail difieren de las noticias USENET en que los diversos mensajes y artículos de discusión se envían a través de e-mail en lugar de a través del soporte de las noticias USENET.

Generalmente, las listas de correo van dirigidas a grupos mas reducidos de personas. Es bastante difícil instalar un nuevo grupo de noticias en USENET, debido a que se requieren periodos de propuesta, discusión y votación. Cualquier administrador de sistemas puede instalar una lista de correo. Debido a que cada lista de correo se mantiene en una computadora, el administrador de

sistemas tiene mas control sobre quien puede estar en la lista y puede tratar mas efectivamente los problemas de los usuarios. Algunas listas de correo, como las que hablan sobre temas de seguridad de computadoras, están restringidas a ciertas personas.

Al igual que con las noticias USENET, existen listas de correo sobre una gran variedad de temas. Regularmente se manda una lista completa de las listas de correo abiertas al publico al grupo de noticias de USENET news.answers.

Las listas de correo se instalan normalmente usando un reflector de correos. Este es una dirección e-mail especial que se instala para reflejar cualquier correo enviado, hacia un grupo de personas. Generalmente hay dos direcciones e-mail asociadas a una lista de correo: la dirección del que mantiene la lista y la dirección de la lista misma. Por ejemplo, imagínese que hay una dirección e-mail para los usuarios de widgets. La dirección email de la lista podía ser algo parecido a

widgets@somewhere.com

Si se envía un mensaje e-mail a esta dirección de lista, se refleja a todas las personas que estén suscritas a la lista. Por convención, las listas de correo de Internet usan una dirección e-mail especial para peticiones administrativas, como subscribirse a la lista. Esta dirección se construye agregando -request al nombre de la lista. Entonces, para la lista de correo de widgets imaginaria, la dirección e-mail administrativa seria

widgets-request@somewhere.com

Todo el correo referido a temas administrativos se debería enviar a la dirección administrativa.

Wide Area Information Servers (WAIS)

WAIS son las siglas de Wide Area Information Servers, que es un sistema de búsqueda de un gran conjunto de bases de datos de información. El termino "wide área" implica ser capaz de usar una gran red, como la de Internet, para llevar a cabo búsquedas usando software de cliente y servidor.

Usando WAIS puede recuperar documentos de texto o multimedia almacenados en bases de datos de todo Internet. Actualmente existen aproximadamente 470 servidores de bases de datos WAIS en Internet. WAIS es similar a Gopher, excepto en que WAIS le realiza la búsqueda.

5.2.6 MOSAIC Y LA WORLD WIDE WEB

La World Wide Web, también conocida por WWW es uno de los desarrollos mas interesantes ocurrido en los últimos años. La Web es un sistema hypermedia interactivo que le conecta a enormes cantidades de información de Internet. Un sistema hypermedia esta compuesto de enlaces entre documentos. Puede acceder a información relacionada solo seleccionando uno de los enlaces. La información se recupera automáticamente sin que tenga que saber donde esta.

5.2.6.1 LOS URL

Se accede a los recursos de la Web por medio de una dirección descriptiva conocida como Uniform Resource Locator, o URL. Puede imaginarse al URL como un puntero hacia un objeto de Internet que no solo le indica donde esta situado el objeto, sino también como se llama y como se accede a el. Todo aquello a lo que se puede acceder con la Web tiene un URL.

La sintaxis de los URL es muy clara. Consiste en un componente que indica que protocolo usar, por ejemplo el http, seguido de dos puntos y de dos diagonales (://). Después contiene el nombre del sistema y la ruta de acceso al archivo que quiere ver. Por ejemplo un URL podría ser

http://www.boutel.com/faq/www_faq.html

a parte a la izquierda de los dos puntos indica el método de acceso para obtener los datos. Hay varios métodos de acceso validos diferentes, que se detallan en la tabla 5.2.

Método	Descripción
http	Protocolo para acceder a la mayoría de las paginas Web. Proporciona enlaces de hypermedia interactivos a las paginas escritas en HTML, el HyperText Markup Language.
wais	Usado para acceder a una ubicación WAIS
gopher	Usado para acceder a un servidor Gopher.
ftp	Proporciona una conexion FTP como usuario anónimo.
telnet	Abre una conexion telnet a un destino.
news	Usado para leer las noticias USENET.

Tabla 5.2 Metodos de acceso validos para los URL

A continuación de las dos barras inclinadas en el URL esta el nombre del sistema y la ruta de acceso al documento que quiere ver o recuperar. Recuerde que este documento puede ser un texto, un documento hypermedia, archivos de sonido, graficos, etc.

5.2.7 DOCUMENTACIÓN ACERCA DE INTERNET

La documentación sobre Internet se compone de varias guía, escritas para facilitar las cosas a los usuarios, y de una serie de documentos, que proporcionan los estándares y las notas de trabajo de la comunidad de investigación de Internet. Esta serie de documentos esta constituida por Request For Comments (RFC), Standars (STD) y For Your Information (FYI). Estas series de documentos suministran información sobre procedimientos, protocolos estándares, historia y futuro de Internet.

.2.7.1 LA SERIE REQUESTS FOR COMMENTS (RFC)

a serie Requests For Comments o RFC, como se la conoce comúnmente, esta formada por ocumentos de trabajo de la comunidad Internet. Estos documentos cubren todos los aspectos de la municación entre computadoras en lo que atañe a Internet. Muchas RFC definen estándares de otocolo, mientras que otras proporcionan información detallada sobre como implementar estos tándares. La mayoría de los estándares actuales se publican como RFC. Las RFC son revisadas nicamente por varios equipos de Internet y por expertos técnicos.

Una vez que se ha asignado un numero de RFC a un documento y este se ha publicado, dicho numero nunca se vuelve a usar al actualizarse o volverse a editar. Los números de los documentos RFC aumentan secuencialmente con cada nueva edición o revisión de estos.

5.2.7.2 LA SERIE STANDARDS (STD)

La serie Standards, también conocida como la STD, es una serie de documentos que define los estándares oficiales de Internet. Los números de la STD solo se asignan a documentos estándar que han completado el proceso de aprobación de estándares de Internet. Los números de la STD se usan para ayudar a identificar las RFC que son la base de los estándares de Internet. A veces se incluye mas de una RFC en una STD y todas tienen el mismo numero de STD. Los números de la STD, al igual que los de la RFC, nunca se vuelven a usar. Si se revisa o actualiza un estándar, se recurrirá a un nuevo numero

La serie STD esta disponible en Internet a través de FTP como usuario anónimo desde diversas ubicaciones, como ftp.internic.net en el directorio /std.

5.2.7.3 LA SERIE FOR YOUR LNFORMATION (FYI)

La serie For Your Information contiene un subconjunto de las RFC diseñadas para proporcionar información general sobre Internet. Estos documentos tienden a ser algo menos técnicos que los de la serie RFC. A cada FYI también se le asigna un numero de RFC. En la Tabla 5.3 se detallan algunos documentos FYI útiles.

La serie FYI esta disponible en Internet a través de FTP como usuario anónimo desde diversas ubicaciones, como ftp.internic.net en el directorio /fyi.

Documento	Descripción
FYI 4	Respuestas a preguntas comunes para nuevos usuarios de Internet
FYI 6	Información sobre el sistema X Windows
FYI 7	Respuestas a preguntas comunes para usuarios de Internet con experiencia
FYI 18	El glosario de los usuarios de Internet
FYI 24	Como utilizar FTP como usuario anónimo

Tabla 5.3 Documentos FYI útiles

5.3 USO DEL CORREO ELECTRÓNICO

5.3.1 DESCRIPCIÓN DE E-MAIL

E-mail es cualquier programa que los usuarios de un sistema usan para enviar y recibir mensajes. Como mínimo, el usuario suministra al programa la dirección del destinatario y el mensaje que quiere enviar. La dirección incluye el nombre de conexion al sistema de la persona que recibirá el correo. Si dicha persona esta en otro sistema en una red, la dirección también incluye una forma de dentificar el sistema informático de destino. El usuario prepara el mensaje mientras usa su rograma e-mail o lo prepara antes usando un editor de texto como el vi.

l uso del correo electrónico tiene las siguientes ventajas:

uede enviar informes, datos y documentos que pueden llegar a su destino en cuestión de segundos minutos.

) tiene que preocuparse si interrumpe a alguien cuando le envía un mensaje, ya que tal errupción es inexistente porque se ocupa de ello el sistema informático.

ede dedicarse a los mensajes recibidos en el momento que le resulte mas conveniente.

Puede enviar correo electrónico a las horas mas intempestivas.

Cuando envía correo electrónico, el sistema informático se encarga de realizar la entrega; esto significa que puede poner el mensaje en una red para que sea entregado en alguna otra ubicación. En este momento, puede considerar que ha enviado el mensaje. Poco después, este llega al sistema del destinatario.

Si el remitente y el destinatario están en el mismo sistema informático, todo este proceso tiene lugar en un sistema. El sistema de correo de la computadora destino verifica que exista el destinatario y el mensaje se agrega a un archivo que guarda todos los e-mails para ese usuario (si no esta conectado a una red, el sistema informático local verifica el destinatario). El archivo de almacenamiento de correo recibe el nombre de buzón del sistema del usuario y tiene la misma denominación que el del usuario que recibe el correo. Por ejemplo, si su nombre de conexion al sistema es usuario1, su buzón del sistema es el archivo denominado usuario1 en el directorio /var/spool/mail. Cuando se ha "entregado" el mensaje en el buzón, se dice que se ha recibido el correo

Su sistema informático le avisa cuando tiene correo. Al leerlo, puede tratarlo mensaje por mensaje y tiene las siguientes opciones:

- Borrar cada mensaje después de haberlo leído o sin preocuparse de leerlo (el uso de e-mail no significa que no vaya a recibir correo basura)
- Guardar mensajes en el buzón del sistema
- Guardar mensajes en el buzón personal

  Guardar mensajes en archivos individuales o carpetas

  Responder directamente al remitente de un mensaje

  Realizar una "respuesta en grupo" a un grupo de usuarios que ha recibido el mismo mensaje

  Remitir correo a otros usuarios

  Imprimir el correo

5.3.2 ENVÍO DE CORREO CON MAIL

Se puede enviar e-mail a una persona, a un grupo de personas o a una lista de correo. Igual que cuando quiere enviar una carta de papel, con el e-mail debe indicar la dirección de los destinatarios. A veces se compondrá o escribirá un mensaje mientras se esta enviando el e-mail, otras se enviara un mensaje preparado y en algunas ocasiones, incluso se puede enviar con el e-mail la salida de un comando o de un programa. Cuando se usa mail o elm, el mensaje que se envíe debe ser un archivo de texto, es decir, un archivo ASCII.

El Simple Mail Transport Protocol (SMTP) se usa para transferir correo entre computadoras. Actualmente solo admite archivos ASCII; para enviar un archivo binario mediante e-mail, se tiene que convertir a ASCII usando la utilidad uuencode.

Independientemente de como se prepara el mensaje, el usuario enviara correo usando un comando con el siguiente formato:

mail dirección<Intro>

Estos comandos inician el sistema de correo. Seguidamente, se puede componer el mensaje de correo y enviarlo a la dirección indicada. En esta sintaxis, dirección es la dirección de e-mail de la persona que recibirá el mensaje. Una dirección puede tener varias formas distintas. Si se va a enviar correo a alguien que tiene un ID de conexion en el mismo sistema que se esta utilizando, se escribe dicho ID. Por ejemplo, para enviar email a alguien conectado a el mismo sistema cuyo nombre de conexion es usuario1, se introduce el siguiente comando:

mail usuario1 <Intro>

Si usuario1 esta en otro sistema al que puede acceder a través de alguna red o conjunto de redes, debe incluir el nombre por el que se conoce este sistema en la red. Supongamos que usuario1 es el

nombre del usuario en un sistema informático cuyo nombre de red es apples.startup.com. Se puede enviar e-mail introduciendo el siguiente comando:

mail usuario1@apples.startup.com<Intro>

0 se puede usar este otro:

mail apples!usuario1 <Intro>

La forma exacta de la dirección depende del tipo de red que se use y de las convenciones o reglas locales.

CONCLUSIONES

La mayoría de los sistemas operativos son programas comerciales apoyados por grandes compañías de software, si se compra un sistema operativo deberá conformarse con lo que el proveedor ofrece, debido a que no se puede modificar el sistema. Linux es un sistema operativo con características muy singulares: es completamente gratuito y el usuario puede hacer modificaciones o mejoras a su conveniencia. Debido a la funcionalidad y disponibilidad de Linux su popularidad se ha ampliado en todo el mundo. Linux es un sistema operativo que puede ejecutarse en un rango muy amplio de plataformas de hardware, ya que se ejecuta desde computadoras con procesador 386.

Linux es una implementación independiente de POSIX, la cual incluye multitarea real, conectividad TCP/IP lo que permite conectarse a Internet, manejo de memoria virtual, librerías compartidas, carga por demanda, manejó apropiado de memoria, entre otras características que lo hacen consistente con las demás implementaciones de UNIX. Además, Linux puede ser usado para una amplia variedad de propósitos, como interconexión de redes, desarrollo de software ya que dispone de lenguajes como C, C++ y SmallTalk; y plataformas para usuario final con lo que respecta a estos, Linux ofrece editores de texto, hojas de calculo. Por lo cual Linux puede ser considerado como una opción económica de UNIX.

Entre los inconvenientes de Linux se encuentra el que como no depende de ninguna organización comercial nadie se hace responsable de los problemas que surjan al utilizar este sistema operativo. Linux también puede resultar difícil de instalar ya que no funciona con todas las plataformas de hardware y el usuario debe de solucionar estos problemas, programando el sistema operativo.

Linux puede ser una alternativa para empresas y particulares que buscan eliminar costos de licencias de sistemas operativos. En México, empresas como Pemex, CFE y el IMSS están usando este sistema operativo.

Linux además ofrece distintos tipos de seguridad (seguridad física, seguridad mediante contraseñas, seguridad de conexión al sistema seguridad de archivos), lo cual es de gran utilidad para los administradores de sistemas.

Por todo lo anterior, se puede concluir que Linux es una muy buena opción para crear servidores de archivos y servidores Web a muy bajo costo, sin importar las características del equipo con el que se cuente.

Además, hoy en día se espera un gran crecimiento de Linux, ya que muchas empresas dedicadas a la elaboración de software, comienzan a tomar interés en desarrollar herramientas para este sistema operativo.

BIBLIOGRAFIA

- Tacket & Gunter, Utilizando Linux 2ª edición, Prentice Hall, México, 1996.
- Clemente de Blas, PC Guía del usuario, Macrobit, México 1991,
- Milan Milenkovik, Sistemas Operativos, Conceptos y Diseños 2ª edición; Mc Graw Hill, 1994.
- Bart Anderson, Bart Anderson y Harry Henderson; UNIX Comminications and the Internet 3ª edición, Sams Publishing, U.S.A. 1995
- Salim Douba, Networking UNIX, Sams Publishing 1ª edición, U.S.A. 1995
- D. Budgen, Introducción al Sistema Operativo UNIX, Gustavo Gili, Barcelona, 1987.
- James Garoner, Aprendiendo Linux 2ª edición, Prentice Hall, 1995.
- Dee Leblanc, Running a Perfect Internet Site with Linux, QUE, 1996.
- Raphale Finkel, Fundamentos de Sistemas Operativos, Prentice Hall, Madrid, 1990
- Andrew Tanenbaum, Modern Operating Systems, Prentice Hall, New Jersey, 1992.
- Paul Renavo, Introduction to Client/Server Systems 2ª edición, Wiley, E.U.A. 1996
- Luis Joyanes Aguilar, Programación Basic para microcomutadoras, 3ª Edicion, Mc Graw Hill, España, 1992.