1
2E

UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO

FACULTAD DE ESTUDIOS SUPERIORES CUAUTITLAN

ADMINISTRACION BASICA DE LOS SERVICIOS DE RED EN SOLARIS 2.X A TRAVES DE TCP/IP

T E S I S

QUE PARA OBTENER EL TITULO DE

LICENCIADO(A) EN INFORMATICA

P R E S E N T A N

ODILIA CASTILLO ROJAS
GERARDO DE LA ROSA PEREZ
ALFREDO ISRAEL OROS CARBAJAL

ASESORES: ING. JOSE JUAN CONTRERAS ESPINOSA
ING. MARLON ENRIQUE CZERMAK ANDRADE

CUAUTITLAN IZCALLI, EDO. DE MEX. **1995**

UNAM – Dirección General de Bibliotecas Tesis Digitales Restricciones de uso

PAGINACION VARIA

COMPLETA LA INFORMACION

VNIVERSIDAD NACIONAL AVFNMA DE MEXICO

FACULTAD DE ESTUDIOS SUPERIORES CUAUTITLAN
UNIDAD DE LA ADMINISTRACION ESCOLAR
DEPARTAMENTO DE EXAMENES PROFESIONALES

ASUNTO: VOTOS APROBATORIOS

DR. JAIME KELLER TORRES
DIRECTOR DE LA FES-CUAUTITLAN
P R E S E N T E .

AT'N: Ing. Rafael Rodríguez Ceballos
Jefe del Departamento de Exámenes
Profesionales de la F.E.S. - C.

Con base en el art. 28 del Reglamento General de Exámenes, nos permitimos comunicar a usted que revisamos la TESIS TITULADA:

"Administración básica de los servicios de red en Solaris 2.X a través de TCP/IP".

que presenta el pasante: Alfredo Israel Oros Carbajal

con número de cuenta: 8706714-5 para obtener el TITULO de: Licenciado en Informática ; en colaboración con : Odilia Castillo Rojas y Gerardo De la Rosa Pérez

Considerando que dicha tesis reúne los requisitos necesarios para ser discutida en el EXAMEN PROFESIONAL correspondiente, otorgamos nuestro VOTO APROBATORIO.

A T E N T A M E N T E .
"POR MI RAZA HABLARA EL ESPIRITU"
Cuautitlán Izcalli, Edo. de Méx., a 28 de marzo de 1995

PRESIDENTE	Ing. José Juan Contreras Espinosa	31/05/95.
VOCAL	Lic. Araceli Nivón Zaghi.	24-V-95
SECRETARIO	Ing. Felipe Beltrán Trejo	23-may-95
PRIMER SUPLENTE	Lic. Armando Carmona Bonilla	2-06-95
SEGUNDO SUPLENTE	Lic. Carlos Pineda Muñoz	14-06-95

UAE/DEP/VAF/02

UNIVERSIDAD NACIONAL AVENTMA DE MEXICO

FACULTAD DE ESTUDIOS SUPERIORES CUAUTITLAN
UNIDAD DE LA ADMINISTRACION ESCOLAR
DEPARTAMENTO DE EXAMENES PROFESIONALES

ASUNTO: VOTOS APROBATORIOS

DR. JAIME KELLER TORRES
DIRECTOR DE LA FES-CUAUTITLAN
P R E S E N T E .

AT'N: Ing. Rafael Rodriguez Ceballos
Jefe del Departamento de Exámenes
Profesionales de la F.E.S. - C.

Con base en el art. 28 del Reglamento General de Exámenes, nos permitimos comunicar a usted que revisamos la TESIS TITULADA:

"Administración básica de los servicios de red en Solaris 2.X a través de TCP/IP".

que presenta el pasante: Gerardo de la Rosa Pérez

con número de cuenta: 8608853-0 para obtener el TITULO de:

Licenciado en Informática ; en colaboración con :
Odilia Castillo Rojas y Alfredo Israel Oros Carbajal.

Considerando que dicha tesis reúne los requisitos necesarios para ser discutida en el EXAMEN PROFESIONAL correspondiente, otorgamos nuestro VOTO APROBATORIO.

A T E N T A M E N T E .
"POR MI RAZA HABLARA EL ESPIRITU"
Cuautitlán Izcalli, Edo. de Méx., a 28 de marzo de 1995

PRESIDENTE	Ing. José Juan Contreras Espinosa	31/05/95
VOCAL	Lic. María Araceli Nivón Zaghi	24-V-95
SECRETARIO	Ing. Felipe Beltrán Trejo	29-MAY-95
PRIMER SUPLENTE	Lic. Armando Carmona Bonilla	2-06-95
SEGUNDO SUPLENTE	Lic. Carlos Pineda Muñoz	1/06/95

UAE/DEP/VAP/02

FALLA DE ORIGEN

VNIVERSIDAD NACIONAL AVPNOMA DE MEXICO

FACULTAD DE ESTUDIOS SUPERIORES CUAUTITLAN
UNIDAD DE LA ADMINISTRACION ESCOLAR
DEPARTAMENTO DE EXAMENES PROFESIONALES

ASUNTO: VOTOS APROBATORIOS

DR. JAIME KELLER TORRES
DIRECTOR DE LA FES-CUAUTITLAN
P R E S E N T E .

AT'N: Ing. Rafael Rodriguez Ceballos
Jefe del Departamento de Exámenes
Profesionales de la F.E.S. - C.

Con base en el art. 28 del Reglamento General de Exámenes, nos permitimos comunicar a usted que revisamos la TESIS TITULADA: "Administración básica de los servicios de red en Solaris 2.X a través de TCP/IP".

que presenta la pasante: Odilia Castillo Rojas

con número de cuenta: 8709867-7 para obtener el TITULO de: Licenciado en Informática ; en colaboración con : Gerardo De la Rosa Pérez y Alfredo Israel Oros Carbajal

Considerando que dicha tesis reúne los requisitos necesarios para ser discutida en el EXAMEN PROFESIONAL correspondiente, otorgamos nuestro VOTO APROBATORIO.

A T E N T A M E N T E .
"POR MI RAZA HABLARA EL ESPIRITU"
Cuautitlán Izcalli, Edo. de Méx., a 28 de marzo de 1995

PRESIDENTE	Ing. José Juan Contreras Espinosa	31/03/95
VOCAL	Lic. Araceli Nivón Zagui	24-V-95
SECRETARIO	Ing. Felipe Beltran Trejo.	29-MAY-95
PRIMER SUPLENTE	Lic. Armando Carmona Bonilla	2-06-95
SEGUNDO SUPLENTE	Lic. Carlos Pineda Muñoz	14-06-95

UAE/DEP/VAP/02

FALLA DE ORIGEN

AGRADECIMIENTOS

Siempre que he logrado un triunfo en mi vida, recuerdo y agradezco a todos aquellos, que son la parte mas importante en mi vida.

A mis padres

Noé y Zenaida

Gracias por el ejemplo mas puro de libertad, superación, y amor en todo lo hacen, y que me han ayudado a superarme día a día.

A mis hermanos

Humberto, Mayeni, Fredy y Arturo

Gracias por su tiempo, paciencia, comprensión y por ser parte de mi vida.

A mis mejores amigos y compañeros

Gerardo y Alfredo Israel

Gracias por todos los momentos que pasamos juntos, por lo que aprendimos, y sobre todo por unir fuerzas para lograr lo que es hoy es una realidad nuestra tesis, y a sus familias mil gracias.

A nuestro Asesor de tesis

Ing. Marlon Czermak A.

Gracias por su apoyo a lo largo de nuestra carrera, con un afán de ser siempre los mejores en cada paso de nuestra vida profesional, junto con el, a su familia por dejarnos ser parte de su vida.

A nuestro Coasesor

Ing. José Juan Contreras Espinosa y su familia

Por todo el apoyo, para lograr nuestro objetivo.

A mis mejores amigos y compañeros, que siempre están presentes.
ASOCIACION MEDALLAS GABINO BARREDA Y LAZARO CARDENAS, A.C.
Liliana P., Isabel B. , Martha R., Luis O. , Paz R. , Isabel P. , Alberto L. , Lourdes M. , Aida G. , Monica Z. , Claudia Z., Adriana E., Cesar Miguel H. , Ernesto A. , Eri P. , Noemi A. , Rafael M.

Con cariño, respeto y el compromiso de ser siempre mejor

Odilia Castillo Rojas.

Agradezco infinitamente a todas aquellas personas que creen en mi, a mi Madre por todos sus sacrificios, por todo su amor y por darme el mejor regalo que una madre le puede dar a un hijo: vida.
Al Ing. Marlon Czermak por todos sus consejos, por "adoptarme" y por que gracias a sus enseñanzas esto es posible.
Al Ing. Juan Contreras y familia gracias infinitas por todo el apoyo recibido.
A la Asociación de Medallas Gabino Barreda y Lázaro Cárdenas A.C. por todo el tiempo y todos los consejos recibidos.
A mis abues por todos sus cuidados, a mis hermanos Javier (por todos los días difíciles), Oscar y Jazmín (por todo su cariño), a mis tíos: Luis (por sus consejos), a Horacio (por sus vivencias), Víctor (por sus experiencias), Norma (por toda su comprensión), Esther (por toda su ayuda), a todos mis primos a quienes lo único que les puedo recomendar es que: escojan el camino que escojan lo vivan intensamente y convencidos de que no existe vuelta atrás.
A Tony Q. , Ara, Chipi y Mona por ese inmenso amor .
A mi "niña" Ali, gracias infinitas por todo tu apoyo y amor que en momentos fue el único motivo y el único "pretexto" para ser más grande y mejor.
A Odi y a Isrra por todas esas ganas y empuje que tienen.
A mis amigos Gaby, Graciela, Ernesto, Horacio, Rafa por todos los momentos compartidos.
El mar que late y habla con la voz del que perdura resplandeciendo.

Agradecimientos:

A mis padres mucho les agradezco el haberme dado todo su apoyo y cariño tanto profesional como ser humano. Sin su ayuda no sería lo que soy ahora ni tendría lo que tengo hoy, con nada puedo pagar lo que ellos me dieron, no obstante trataré de enorgullecerlos día a día.

A mis abuelos(vitos) por haberme dado tanto cariño y apoyo durante todos estos años. Especialmente ahora en mi vida de casado. Sin lugar a dudas ustedes guardan un lugar muy especial dentro de mi persona.

A mi esposa por motivarme y apoyarme incondicionalmente para el logro tanto de nuestras metas comunes como de las mías propias.

A mis hermanos y a mi hermana por ayudarme a solucionar algunos problemas y darme su cariño.

A mis amigos: Rafael Mondragón Trejo, Odilia Castillo Rojas, Gerardo De la Rosa Pérez y Silvia Dino Zepeda, gracias por los momentos de ayuda mutua y de diversión que pasamos juntos.

Muy especialmente al Ing. Marlon Czermak Andrade y familia por habernos proporcionado su tiempo, privacidad y esfuerzo para que el día de hoy nos podamos titular.

Al Ing. José Juan Conteras por permitirnos alcanzar con mayor facilidad nuestros objetivos de una manera franca y desinteresada y por habernos brindado su amistad.

Título:

Administración básica de los servicios de red en Solaris 2.X a través de TCP/IP.

Justificación del tema:

La Informática como carrera multidisciplinaria requiere de la conjunción de diversos tipos de conocimientos para la resolución de problemas.

Es común encontrar en el campo laboral personal encargado de redes que no conoce posee la preparación necesaria para llevar a cabo dichas funciones. Estas deficiencias generan una gran problemática en los sistemas que administran, por ejemplo: pérdida de información al realizar respaldos de forma incorrecta, mala planeación del crecimiento del área de sistemas, desaprovechamiento de los recursos de que se dispone, generación de tiempos muertos debidos a mal funcionamiento del sistema, entre muchos otros; como es de suponerse uno de los principales objetivos de una organización al implementar un sistema de cómputo es el disminuir costos y en general la automatización de las diversas funciones que se realicen, es por eso que en base a los problemas anteriormente descritos y a las necesidades actuales es un requisito indispensable para los profesionistas en el área informática el contar con una sólida preparación.

Dentro de las diferentes áreas que existen dentro de la informática centraremos nuestra atención hacia las redes de cómputo y en específico a aquellas que utilizan como sistema operativo a Solaris, y como herramientas auxiliares a los

protocolos de comunicación a TCP/IP y al Protocolo X, debido a que es una de las versiones de UNIX con mayor base instalada a nivel mundial, y los protocolos TCP/IP y X han sido adoptados por la mayor parte de los fabricantes de software y de hardware.

Si bien los tres temas pueden ser tratados cada uno de ellos como un trabajo de tesis, existen varias razones por las cuales decidimos conjuntar nuestra experiencia en las tres "áreas" y exponerla como una sola por las siguientes razones:

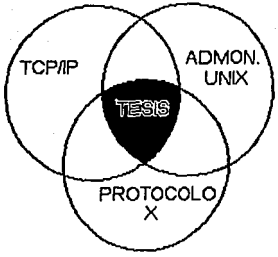
a) Si se tuviera un ambiente de red en la cual convivieran varios servidores de la misma plataforma, ¿Cómo sería posible compartir periféricos, servicios de impresión e información desde cualquiera de los nodos dentro de una red?. La respuesta se encuentra bajo los dominios de TCP\IP como parte de una herramienta de administración.

b) ¿Sería posible el tener varias aplicaciones que se desplegarán en una sola terminal simultáneamente?. La respuesta la hayamos bajo el protocolo X, conjuntamente con el protocolo de comunicación TCP\IP.

c) Otra pregunta sería ¿Es posible administrar una red, desde cualquier nodo de la misma?. La respuesta a esta pregunta se encuentra bajo los tres nichos que se exponen aquí.

Y así como este tipo de preguntas se pueden enumerar muchos problemas que suceden en una red UNIX.

Por lo que no es posible desligarse de estos tres nichos si se desea encontrar una solución integral y óptima que ataque los problemas de una forma clara y no redundante.



El área sombreada es lo que nosotros conceptualizamos como la tesis de Administración de los servicios de red con TCP/IP y el Protocolo X.

Otro de los aspectos importantes es el compromiso que poseemos con la Universidad Nacional Autónoma de México para la formación integral de profesionistas que sean de utilidad a la sociedad, es preciso que los estudiantes tengan el acceso a estos conocimientos ya que dentro de nuestra universidad, dichas especialidades no se imparten como cátedras dentro de las carreras relacionadas al área de informática, sino que tan sólo se pueden adquirir dichos conocimientos al tomar cursos extracurriculares. Es por ello, que concluímos que si reunimos los conocimientos aislados de cada uno de nosotros y tratamos de complementarlos entre sí, éstos podrán servir como una útil guía para la formación académica y profesional de cualquier estudiante del área.

Objetivos:

Objetivo General:

Dar a conocer los aspectos esenciales de la administración del sistema operativo Solaris 2.X, el protocolo de comunicaciones TCP/IP y del protocolo X; para que de esta manera se pueda llevar a cabo la administración de servicios de red.

Objetivos Particulares:

Proporcionar al futuro profesionista en el área de la informática los conocimientos necesarios que le permitan introducirse en la administración de los servicios de red en una plataforma SUN.

Brindar al estudiante del área de sistemas conocimientos - tanto conceptuales, como prácticos- en torno a la administración de los servicios de red en UNIX.

Aumentar el desempeño del sistema operativo de la red, a partir de la reconfiguración del sistema operativo Solaris 2.X.

Aprovechar el equipo de cómputo e instalaciones, a través de la utilización de los procolos de comunicación.

Objetivos Específicos:

Establecer los aspectos teóricos-prácticos que permitan al profesional o pasante de licenciaturas relacionadas con sistemas de cómputo, el ampliar su campo de soluciones, basándose en UNIX, TCP/IP y el Protocolo X.

Determinar los aspectos que deben de ser tomados en cuenta antes de instalar una red de cómputo basada en UNIX.

Mostrar la forma en la cual se puede implementar una comunicación entre sistemas UNIX vía módem.

Demostrar que a través del protocolo X se pueden correr en una misma terminal aplicaciones que se encuentren corriendo en diferentes servidores.

Hipótesis:

A través de TCP\IP se pueden compartir periféricos, servicios de impresión e información desde cualquiera de los nodos dentro de una red.

Si se utiliza el protocolo X, entonces será posible tener varias aplicaciones desplegándose simultaneamente en una terminal X.

Se puede realizar la administración de una red bajo Solaris 2.X desde cualquier punto de la misma, si se auxilia de los protocolos TCP/IP o X.

Si se implementa NFS en la red, se podrá disminuir la duplicidad de información.

Si se reconfigura el Kernel de Solaris 2.X en base a las necesidades de trabajo de los usuarios, entonces puede mejorarse el desempeño del sistema.

Si se configuran equipos como servidores de impresión, entonces se pueden reducir los costos de operación dentro de las empresas, al utilizar de forma más eficiente el equipo con que se cuenta.

INDICE

13. Apéndices

14. Glosario

15. Conclusiones

16. Bibliografía

Introducción:

En la actualidad, uno de los problemas más grandes a los cuales se tienen que enfrentar los profesionistas en informática es el escaso o nulo contacto con las tecnologías de punta prevalencientes en el "mundo real", es por ello que, la intención del presente trabajo es proporcionar a los estudiantes y profesionistas de las carreras relacionadas con el estudio de la informática los conocimientos técnicos básicos acerca de la administración del sistema operativo Solaris de SUN Microsystems, del Protocolo de Comunicaciones TCP/IP y del Protocolo X, para que de esta forma puedan tener herramientas adicionales a su formación académica y sobre todo que le permitan poderse integrar más rápidamente al sector productivo de esta área.

Como se planteó anteriormente, es preciso mencionar que aún parte de las personas que actualmente se encuentran trabajando en el área de sistemas, carecen de bases sólidas que sustenten las tareas que están realizando. Esta problemática fue percibida durante el tiempo que estuvimos trabajando dentro del área de capacitación de la empresa SilVaTech, ya que ahí tuvimos la posibilidad de conocer a personas responsables de las áreas de cómputo de sus respectivas empresas así como su nivel de conocimientos acerca de sistemas. En base a ello surgió la idea de crear una tesis que pudiera apoyar a la formación de los profesionistas dentro del área de informática.

La presente tesis está dirigida principalmente a estudiantes que se encuentren en los últimos semestres de carreras dentro del área de informática, y profesionistas que reúnan los siguientes requisitos:

- Conocimientos básicos de redes.
- Conocimiento de UNIX a nivel básico.
- Conocimientos básicos de alguna interfaz gráfica.

En el desarrollo de la tesis se irán tocando los principales aspectos teórico-prácticos con los que deberá de contar el futuro profesionista

en el área, de una manera gradual y progresiva, es decir, se darán las bases de qué es un sistema operativo y de sus principales características, de la administración del sistema operativo UNIX (Solaris 2.X), del protocolo de comunicaciones TCP/IP y del protocolo X.

Cabe mencionar que los primeros cinco capítulos tienen un carácter meramente introductorio y los conceptos manejados en ellos son explicados más ampliamente en los capítulos posteriores.

La aportación de esta tesis se encuentra plasmada en la sección de apéndices, mismos que hacen una breve reseña del problema, los comandos necesarios y el procedimiento para su solución.

Notas aclaratorias:

La presente sección tiene por objeto plantear ciertas consideraciones que deberán ser tomadas para la interpretación de esta tesis.

En primera instancia es preciso mencionar que el leguaje a utilizar durante el desarrollo de la tesis será técnico llano.

Con respecto a la notación y simbología utilizadas para representar los diferentes tópicos a exponer, serán utilizadas las siguientes:

Concepto	Fuente
Términos técnicos	*Arial itálica*
Comandos	**Arial bold**
Parámetros de comandos	*Arial itálica*
Salida de los comandos	***Arial itálica-bold***

Por otra parte, en el caso de que se requiera de alguna ilustración o diagrama se indicará el correspondiente entre paréntesis. Ejemplo: (Ver figura X.1).

Con respecto al formato de los títulos correspondientes a los capítulos, subcapítulos y apartados será la siguiente:

Capítulos, la fuente de la letra será Arial de 20 puntos en bold.

Capítulos

Subcapítulos, la fuente de la letra será Arial de 18 puntos en bold.

Subcapítulos

Apartados, la fuente de la letra será Arial de 16 puntos en bold.

Apartados

La clasificación que será utilizada será decimal.

Las indicaciones a nota de pié de página deberán ser superíndices a lado del párrafo que hace referencia a algún otro texto, la fuente será Arial de 10 puntos. La orden será en base a una numeración consecutiva por cada capítulo, en forma decimal, siendo el primer dígito el número del capítulo y el segundo dígito el número de nota. Por ejemplo:

Unix es un sistema operativo multiusuario de tiempo compartido...[X.1]

Las notas de pié de página o aparato crítico deberán ser puestas al final de cada hoja, la fuente utilizada será Arial de 10 puntos. Por ejemplo:

X.1 Levin, Revista Ciencia y Desarrollo vol. XIV, núm. 83, pp. 169-178

Sun. Cuando se haga referencia a "Sun", lo que deberemos entender es que se habla de la compañía Sun Microsystems, Inc.

Sun OS. Cuando se utilice esta palabra se hará referencia al sistema operativo de Sun (Sun Operating System).

Marcas registradas

1. ¿Qué es un sistema operativo?

PAGINACION VARIA

COMPLETA LA INFORMACION

1. ¿Qué es un sistema operativo?

1.1 Definiciones

Un sistema operativo es un programa que maneja todos los recursos de la computadora, incluyendo la unidad de control de proceso, memoria y periféricos de almacenamiento externo como discos y cintas. El sistema operativo también controla el uso de terminales e impresoras. [1.1]

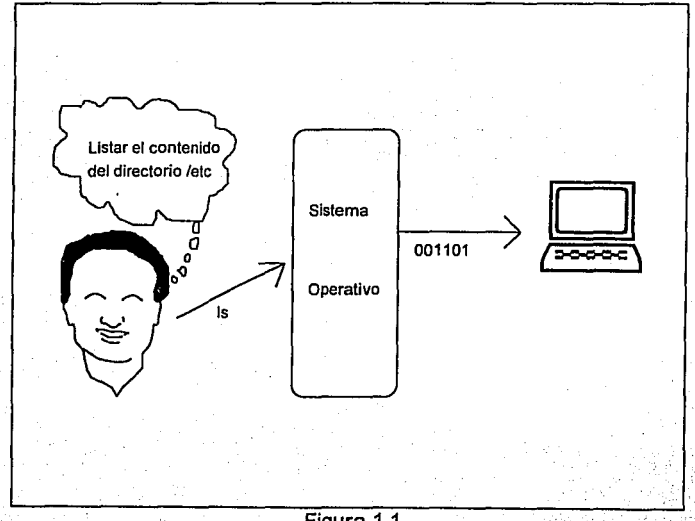


Figura 1.1

1.1 SilVaTech, Manual de Fundamentos de Unix, p. 2

Visto de otra forma, el sistema operativo es una interface entre las instrucciones que pueden ser comprendidas fácilmente por el hombre y las instrucciones que pueden ser entendidas por la computadora. (Ver figura 1.1)

"Por sí sola, una computadora es un conjunto de metal, silicio, equipo de comunicaciones y material magnético. Son los programas de aplicaciones los que hacen que una computadora sea útil a sus usuarios. Las aplicaciones pueden variar desde el procesamiento de textos, pasando por la escritura de compiladores hasta la generación de nuevos sistemas operativos.

Para explotar las posibilidades del hardware de una computadora en comunicaciones, almacenamiento de datos y tratamiento de la información, el software de aplicación requiere de algún tipo de supervisor que pueda manejar los detalles de la gestión de los recursos del software, del acceso a los ficheros y de la interacción con los usuarios. Estas funciones de supervisión son el trabajo del sistema operativo" [1.2]

La función general de un sistema operativo es controlar y dirigir la operación de la computadora, de forma tal que presente una imagen monolítica y virtual (en contraposición con real, electrónica o ingenieril) ante los usuarios del sistema de cómputo, estaremos de acuerdo entonces en que dicho sistema es tan importante como las facilidades físicas y electrónicas de dicho equipo. [1.3]

1.2 Morgan y McGilton, Introducción al Unix Sistema V, p. 2
1.3 Levin, Revista Ciencia y Desarrollo vol. XIV, núm. 83, p. 169

1.2 Características de los de sistemas operativos.

Lo que se espera de un sistema operativo es que sea capaz de atender la operación concurrente de múltiples pedidos de atención por parte de los procesos que están ejecutándose en la computadora; que pueda mantener toda la operación bajo control sin perder detalle alguno ni permitir que los procesos interfieran entre sí; que logre un aprovechamiento óptimo de los recursos físicos de la máquina (procesador, memoria, periféricos,) y, por último, que haga esto de forma silenciosa y eficiente. [1.4]

Algunas de las principales características de los sistemas operativos son:

Interactividad. Es una forma de operación de los sistemas, en la que hay una respuesta a las instrucciones del operador cuando éstas entran. Las instrucciones pueden presentarse por medio de un dispositivo de entrada tal como un teclado o un lápiz foto sensible y el efecto se observa con la suficiente rapidez como para que el operador pueda trabajar casi continuamente. [1.5]

Monousuario. Es cuando solamente un usuario puede hacer uso de las facilidades del equipo de cómputo, tales como ejecución de programas, uso de periféricos y respaldo de información, entre otros.

Multiusuario. Significa que varios usuarios están conectados a una sola computadora y hacen uso de los recursos -como impresoras, tiempo de procesamiento, acceso a disco, respaldos, entre otros-, de forma simultánea o concurrente. Este tipo de sistemas requieren de mecanismos de seguridad muy complejos, ya que suponen el acceso de varias personas al sistema, con los consecuentes problemas de confidencialidad y seguridad de la información, de tiempos de acceso al procesador y de colas de espera para el uso de los diferentes dispositivos existentes. El aspecto multiusuario es una consecuencia natural de la característica multitarea.

1.4 Levin, op.cit, p. 169
1.5 Díaz, Diccionario de Informática, p. 82

Multitarea. Esto significa que el sistema puede realizar varias tareas u órdenes, de manera simultánea. La característica multitarea significa que se le pueden dar al sistema una o más tareas para hacer en modo de prioridad subordinada y mientras se puede seguir haciendo otras cosas distintas, sin tener que esperar a que las otras tareas terminen.

Multiprocesamiento. Esto significa que el sistema puede aprovechar al máximo el uso de los equipos que posean varios procesadores, al repartir -en algunos equipos de forma simétrica- la carga de trabajo entre todos ellos.

Portabilidad. Es la facilidad de poder trasladar un programa de una plataforma a otra, con tan sólo algunas pequeñas modificaciones en su estructura -prácticamente sin ningún problema-. Es decir, que las aplicaciones que se desarrollen en un sistema puedan ser utilizadas en algún otro, debido a que éstas no fueron creadas con la intención de que corrieran únicamente en él. [1.6]

Estandarización. Se refiere a una serie de definiciones que tienen por objeto mantener dentro de una línea a los diferentes creadores de software, hardware o de alguna otra rama en particular. Los estándares no pretenden imponer algo en específico, sino que se basan en un concenso de las características más útiles y comunes de un fenómeno en particular, de manera que se puedan generar al rededor de éstos nuevas implementaciones. [1.7]

1.3 Ejemplos de sistemas operativos

Dentro de los sistemas operativos más conocidos e importantes dentro del mundo de la computación podemos mencionar:

1.6 Tanenbaum, Sistemas Operativos Modernos, pp. 302-303
1.7 Hewllet-Packard, Fundamentals of the UNIX Operating System, p. 1-16,1-17

MS-DOS (MicroSoft Disk Operating System). El cual es quizás el sistema operativo para computadoras personales de mayor demanda y difusión a nivel mundial. Como sus principales características están: el ser monousuario y monotarea. Trabaja sobre *PC's*.

Novell. Es un sistema operativo de red y multiusuario. En la actualidad es uno de los sistemas operativos más afamados de la actualidad. Básicamente trabaja sobre *PC's*.

UNIX. Es un sistema operativo creado en los Laboratorios Bell de New Jersey, E.U. En la actualidad *UNIX* se considera un estándar para computadoras multiusuario, y ha sido adoptado por gran cantidad de máquinas. Entre sus principales características se encuentran las siguientes: permite el multiprocesamiento, la multitarea, el procesamiento no interactivo, emplea el manejo dinámico de memoria, tiene capacidad de interconexión entre procesos y garantiza un alto grado de portabilidad. Funciona en plataformas de PC's, estaciones de trabajo, minis e incluso en supercomputadoras.

Windows NT. Es otro sistema operativo de red -creado por Microsoft-, con características de multiusuario y de simulación de multitarea. Incluye paquetes adicionales que le permiten interactuar con otras redes vía TCP/IP. Opera sobre PC's (486 en adelante) y estaciones de trabajo.

También están los sistemas operativos de red *Netware* (Novell), *Lantastic* (Artisoft), *Lan Manager*, *MVS* (Digital Equipment), *CMS* (IBM); por citar tan sólo algunos otros ejemplos de sistemas operativos, no se profundizará más en este tema, debido principalmente a que tiene un carácter meramente introductorio y a que el objetivo de esta tesis no es el estudio de los diferentes sistemas operativos.

2. ¿Qué es una red?

2. ¿Qué es una red?

2.1 Definiciones

Una red son al menos dos computadoras que comparten recursos, tales como discos duros, impresoras, dispositivos de almacenamiento, programas.

Una red es un grupo de ordenadores y terminales, en general, interconectadas a través de uno o varios caminos o medios de transmisión. Las redes tienen una finalidad concreta: transferir e intercambiar datos entre ordenadores y terminales. [2.1]

Una red local es un sistema de comunicaciones de datos que permite a un número de dispositivos independientes comunicarse entre sí. [2.2]

2.2 Tipos de redes

2.2.1 ¿Que es una *LAN* (Local Area Network)?

Una Red de Área Local (Local Area Network) es tecnología utilizada para interconectar dispositivos en una misma área física, de tamaño usualmente no mayor a un edificio de oficinas estándar (Ver figura 1).

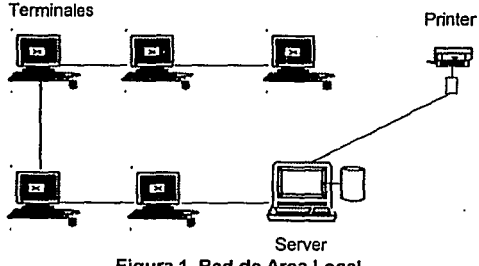


Figura 1. Red de Area Local

2.1 Black, Redes de Computadoras, p. 1
2.2 Comité 802 del IEEE

Este tipo de red se ubica en una sola entidad, edificio o campus. Un atributo claro de una *LAN* es la conectividad, que es la posibilidad de comunicarse desde cualquier punto dado dentro de la red a otro.

Las *LAN* son importantes para muchas organizaciones de tamaño pequeño, porque son la ruta a seguir en un entorno de computación multiusuarios distribuído, capaz de comenzar en forma modesta, pero también de extenderse a medida que las necesidades de la organización así lo exijan.

Su extensión varía de unos cuantos metros hasta 50 kilómetros.

2.2.2 ¿Qué es una red WAN (Wide Area Netwrok)?

Es la unión de dos o más *LAN*'s geográficamente distantes (Ver figura 2). Las redes globales, algunas veces llamadas redes de área vasta, tienen una o más computadoras nodo centrales para la operación de la red.

El nodo central suele ser un *mainframe* o una *minicomputadora* de tiempo compartido. Al construir o rediseñar una red **WAN**, a menudo no se toman en cuenta a máquinas específicas, sino que se fabrican en torno a aspectos de conectividad globales. Esta es la realidad del concepto de interconexión de redes que se ha desarrollado en el modelo Open Systems Interconections (*OSI*) y los estándares de *TCP/IP* (OSI y TCP/IP serán explicados en capítulos siguientes).

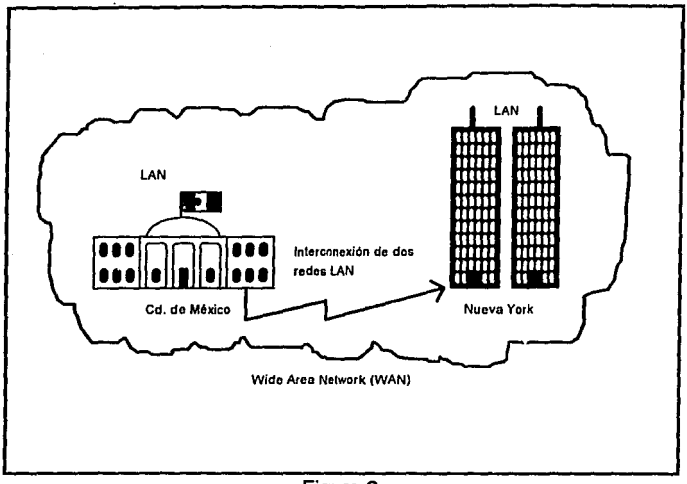

LAN
LAN
Interconexión de dos redes LAN
Cd. de México
Nueva York
Wide Area Network (WAN)

Figura 2.

3. Modelo OSI

3. Modelo OSI

3.1 Definiciones

La mayoría de las redes se organizan en una serie de *capas* o *niveles*, con objeto de reducir la complejidad de su diseño. Cada una de ellas se construye sobre su predecesora. El número de *capas*, el nombre. contenido y función de cada una varían de una red a otra. Sin embargo, en cualquier red, el propósito de cada capa es ofrecer ciertos servicios a las *capas* superiores, liberándolas del conocimiento detallado sobre cómo se realizan dichos servicios. 3.1

Cuando la capa *n* conversa con la capa *n* de otra máquina, usan una serie de convenciones y reglas, a ello se le conoce como *protocolo* de la capa *n*.

Al conjunto de *capas* y *protocolos* se le denomina *arquitectura de red.* Las especificaciones de está deberán contener la información suficiente que permita escribir un programa o construir hardware.

Haciendo una analogía tenemos la siguiente comparación:

Modelo	=	Arquitectura
Nivel	=	Función
Protocolo	=	Reglas

3.1 Arquitectura de redes, Andrew S. Tanenbaum

3.2 Modelo de referencia *ISO/OSI*

La Organización Internacional para la Estandarización (International Organization for Standarization, *ISO*), tomó la rienda en 1977 con la finalidad de crear un modelo de normalización de varios protocolo para ofrecer una interoperabilidad y acceso universal de diferentes vendedores en sus productos.

ISO diseñó un estándar al que llamó Interconexión de Sistemas Abiertos (Open Systems Interconection - *OSI)*, con la capacidad de comunicarse entre diversos equipos de cómputo, compartiendo información a través de diferentes organismos y departamentos internacionales.

El modelo *OSI* se fundamenta en siete capas de operación.
Los principios aplicados para el establecimiento de las siete capas fueron los siguientes:

1. Una *capa* se creará en situaciones donde se necesite un nivel diferente de abstracción.

2. Cada *capa* deberá efectuar una operación bien definida.

3. La función que realizará cada *capa* deberá seleccionarse con la intención de definir protocolos normalizados internacionalmente.

4. Los límites de las *capas* deberán seleccionarse tomando en cuenta la minimización del flujo de información a través de las interfaces.

5. El número de *capas* deberá ser lo suficientemente grande para que funciones diferentes no tengan que ponerse juntas en la misma capa, y por otra parte, deberá ser lo suficientemente pequeño para que su arquitectura no llegue a ser difícil de manejar.

El modelo *OSI* es una estructura que divide funcionalmente las comunicaciones de datos distribuidas, a través de siete capas (Ver figura 3.1).

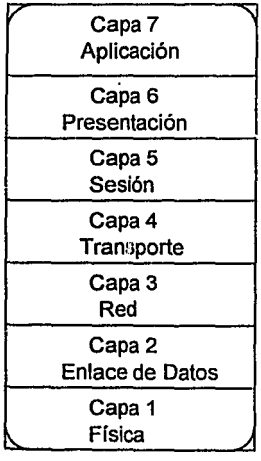


Figura 3.1

Supongamos que tenemos dos máquinas que se comunicarán entre sí usando una arquitectura OSI, cada una de ellas tiene un modelo OSI dividido en siete capas, cada capa de comunicará con su igual en la otra máquina a esto se le denomina *procesos pares.*

En realidad no existe una transferencia de datos entre capas, sino que cada capa envía la información con una etiqueta propia y la pasa a la capa siguiente y así sucesivamente hasta que llega a la capa física que es donde en realidad se envía a la máquina remota el proceso se invierte en la máquina remota de la capa 1 a la 7.

Cabe aclarar que el modelo OSI, por sí mismo, no es una arquitectura de red, dado que no especifica, en forma exacta, los servicios y protocolos que se utilizarán en cada una de las capas. Sólo indica lo que cada capa deberá hacer, es por eso que en muchos textos se conoce como modelo de referencia OSI/ISO.

4. Protocolo de comunicación TCP/IP

4. Protocolo de comunicación TCP/IP

4.1 Definiciones

TCP/IP es uno de los programas que se incluyen en UNIX para comunicación de datos. Este juega un importante papel como el software líder en comunicaciones para las redes locales basadas en UNIX.

El nombre de TCP/IP se refiere a un conjunto de protocolos de comunicación de datos. TCP/IP toma su nombre de dos protocolos que conforman este nicho: Transmission Control Protocol e Internet Protocol. sin embargo existen mas protocolos bajo la etiqueta de TCP/IP; ciertamente TCP e IP son dos de los más importantes.

TCP/IP no es una serie de protocolos que se hayan realizado recientemente. En 1969 la DARPA (Defense Advanced Research Proyects Agency) fundó un proyecto de investigación y desarrollo que creará un software de intercambio de datos a través de una red. Está red llamada ARPANET, fue construida para estudiar técnicas que proporcionaran robustez, confiabilidad e independencia de los constructores de hardware. Muchas de las técnicas modernas de comunicación de datos fueron desarrolladas en ARPANET.

Debido al éxito de ARPANET muchas organizaciones se empezaron a conectar para poder usar las comunicaciones de datos. En 1975 ARPANET fue convertida de una red experimental a una red operacional, y la responsabilidad de la administración de la red fue proporcionada por la DCA (*Defense Communications Agency*), sin embargo el crecimiento de ARPANET no se pudo detener por que se empezó a usar como una red operacional. Los protocolos básicos de TCP/IP fueron desarrollados después de que ARPANET fuera operacional. El protocolo TCP/IP fue adoptado por MIL STD (*Military Standards*) en 1983, y todos los *hosts* (ver glosario para definición), conectados a la red fueron requeridos para usar el nuevo protocolo.

Para cumplir este requerimiento, DARPA fundó Bold, Beranek y Newman (BBN) para implementar TCP/IP en UNIX Berkeley (BSD), esto fue el comienzo del "matrimonio" entre UNIX y TCP/IP.

Por este tiempo TCP/IP fue adoptado como estándar y el termino *Internet* se volvió de uso común. En 1983, la vieja ARPANET fue dividida en MILNET y una nueva y pequeña ARPANET. El termino *Internet* se uso para referir a la red entera MILNET más ARPANET. En 1990, ARPANET formalmente dejo de existir. Sin embargo *Internet* conecta muchas redes alrededor del mundo.

Una de las confusiones más comunes es el termino *internet*. Originalmente el nombre solamente era usado para referir al protocolo *internet* (perteneciente a TCP/IP). Ahora "*i*"*nternet* es un nombre

genérico usado para referenciar una colección de redes físicamente separadas, interconectadas bajo un protocolo en común, para formar una sola red lógica. "*I"nternet* es una colección a lo ancho del mundo de redes interconectadas, que usan el protocolo *IP* (*Internet Protocol*) para ligar varías redes dentro de una sola red lógica.

Para los fines de está tesis "*internet*" e "*Internet*" se referirán a la interconexión de redes por TCP/IP (salvo en las ocasiones que se indiquen).

4.2 Servicios de TCP/IP

Los servicios "tradicionales" de TCP/IP son soportados por los protocolos adecuados que se describen brevemente a continuación (los protocolos que se presentan a continuación son algunos de los que componen a TCP/IP los restantes y estos mismos son tratados en profundidad en el tema 9).

Estos *protocolos* son:

El protocolo de transferencia de archivos (*FTP ; File Transfer Protocol*), que hace posible la transferencia de archivos de una computadora a cualquier otra computadora (bajo TCP/IP o con alguna herramienta compatible con este).

El protocolo de terminales de red (*Telnet*) ofrece un medio para permitir a un usuario, ingresar a cualquier otra computadora de red.

El protocolo simple de transferencia de correspondencia (*SMTP*) permite a los usuarios enviar mensajes entre si.
Cada uno de los servicios implícitos en estos protocolos deben estar presentes en general en cualquier implantación de TCP/IP; aunque SMTP no es soportado en todos los casos por sistemas de micro computadoras.

Otros servicios que se ofrecen dentro del campo de acción de TCP/IP son:

Sistemas de archivos para redes
Impresión distante
Ejecución distante.
Servidores de nombres.
Servidores de terminales.
Sistemas de ventanas orientados a redes.

4.3 Capas de TCP/IP

El Modelo TCP/IP describe su marco de comunicaciones usando cuatro capas. Una breve descripción de cada capa es la siguiente :

Aplicación

Esta capa incluye todos los procesos que son usados para que la capa de transporte envíe los datos. En esta capa se definen muchos protocolos de aplicación:

Los más ampliamente conocidos son *TELNET, FTP y SMTP.* (Los protocolos de esta capa son explicados más ampliamente en el capítulo 9)

Transporte

El nombre completo de esta capa es: *Host-to-Host Transport Layer* aunque su nombre corto y más conocido es el de *Transporte.* Existen dos protocolos que se manejan en esta capa *Transmission Control Protocol (TCP) y User Datagram Protocol (UDP). TCP* prove una conexión más segura a través de un servicio con detección de errores punto-a-punto y corrección de los mismos. *UDP* provee un servicio sin conexión, es decir no establece una conexión punto-a-punto. (Estos dos protocolos se definen más ampliamente en el capítulo 9).

Internet

El protocolo *Internet* es el corazon de *TCP/IP*. Todos los protocolos en las capas superiores son "pasados" por *IP*.

El protocolo *Internet* es el bloque sobre el cual de contruyó la red Internet. Dentro de las funciones de *IP* se enumeran las siguientes:

- Define Datagramas. Que es la unidad básica de trasmisión de Internet.
- Define el esquema de direccionamiento de Internet.
- Mueve datos entre la capa de red y la capa de transporte.
- Rutea Datagramas a hosts remotos.
- Mantiene el performance de fragmentación y re-ensamblaje de datagramas.

(Se agregan más detalles en el capítulo 9).

Acceso a Red

Consiste de rutinas para accesar para accesar el medio físico de las redes (no es objetivo de esta tesis el definir los medios y los formatos que existen para conectar redes, ni las topologías existentes).
El acceso a las capas de red (Red, Enlace de Datos y Física) es de alguna manera ignorada por los usuarios. El diseño de *TCP/IP* oculta el funcionamiento de las capas inferiores, es por eso que los protocolos mas conocidos son *TCP, UDP, IP, etc.*

Una de las enormes ventajas que proporciona *TCP/IP* es que cuando aparece una nueva tecnología de hardware, los nuevos protocolos de *acceso a red* se desarrollan para que las redes *TCP/IP* puedan usar el nuevo hardware.

4.4 Comparación de OSI vs TCP/IP

El siguiente diagrama compara el Modelo de Referencia *ISO/OSI* con el Modelo *TCP/IP* (ver figura 4.1).

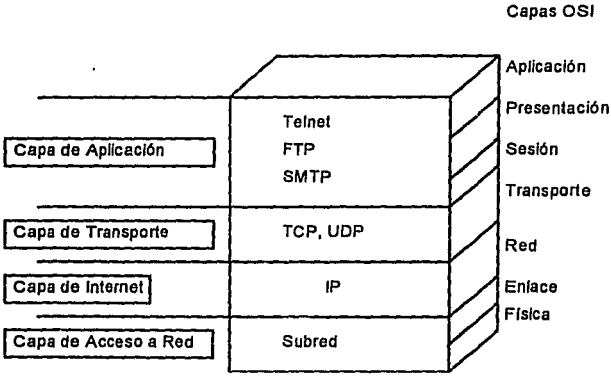


figura 4.1

No existe realmente realmente una descripción universal para definir *TCP/IP* por capas, generalmente se le representa con menos capas que al modelo OSI, el modelo de cuatro capas que se muestra en la figura 4.1 se basa en tres capas (Aplicación, Transporte e Internet). [4.1]

4.1 Overview of TCP/IP; TCP/IP Network Administration p.p.9.

4.5 Comunicación Puerto a Puerto

Cuando hay intercambio de datos entre sistemas utilizando el modelo *TCP/IP*, se tiene comunicación puerto a puerto. Comunicación puerto a puerto es la habilidad de una capa en particular de comunicarse con la capa correspondiente en otro host.

En cada nivel, el mensaje se encapsula y se le incluye un encabezado de información sobre el nivel de protocolo correspondiente. Esta información es clave en la comunicación puerto a puerto y se usa para desencapsular y direccionar el mensaje a la aplicación apropiada. La encapsulación de información se discute en los capítulos referentes a *TCP/IP*.

5. Protocolo X

5 PROTOCOLO X

5.1 Definición

Actualmente en el mundo de la informática, como en muchos otros ambientes, estamos en un período de transición de una moda hacia otra. Ahora todo es *Cliente-Servidor.* Que para este tema lo veremos como un modelo de construcción de programas en el cuál intervienen diversos componentes que definimos más adelante (Ver tema 10.2).

El *protocolo X* es la verdadera definición del *sistema de ventanas X,* y cualquier código en cualquier lenguaje que lo utilice es una implementación real de *X.*

El *protocolo* esta diseñado para comunicar toda la información necesaria para operar un *sistema de ventanas.* El *protocolo* se utiliza para envío de comandos de ventanas y gráficas. (Ver figura 5.1).

5.2 Conceptos

5.2.1 Sistema de ventanas X (X Window System)

Sistema de ventanas en *red*, basado en el *Protocolo X*, para terminales de mapeo de bits.

El *sistema de ventanas X* es la descripción sobre la que se debe basar cualquier *cliente* escrito para X, incluyendo un manejador de ventanas.

5.2.2 Terminal X (*X Terminal)*

Unidad que despliega aplicaciones de múltiples *hosts, (servidores de aplicaciones)* en una *red* usando un sólo *sistema de ventanas.*

Su *hardware* generalmente incluye memoria propia y un procesador local para acelerar los procesos de despliegue y liberar la carga del servidor.

5.2.3 Servidor de X (*X Server)*

Software que corre en la *estación de trabajo* o en la *terminal de despliegue* que :

* Acepta requisiciones de *clientes* en *Protocolo X* y las convierte en instrucciones que manejan los dispositivos de entrada y salida.

* Convierte eventos de *estaciones de trabajo* y despliegue en *protocolo X* y los pasa al *servidor.*

* Controla la pantalla de despliegue, teclado, mouse y cualquier otro dispositivo de entrada conectado a la *estación de trabajo.*

5.2.4 Librerías de X (*Xlib)*

Colección de subrutinas usadas por las aplicaciones para comunicarse con el *servidor de X. Xlib* comprende todas las utilerías implementadas para la versión de X a la fecha, modificadas y actualizadas. El *sistema de ventanas X* sigue la implementación de estas librerías.

Normalmente, los *clientes* implementan el *protocolo X* usando una librería de programación que realiza la interface a una sola *capa* de red inferior, típicamente en *TCP/IP.* (Ver figura 5.2).

5.2.5 Servidor de aplicaciones

Cualquier máquina en la red donde resida una aplicación que se despliegue en ella misma o cualquier otro servidor de despliegue.

5.3 El *sistema de ventanas X*

El *sistema de ventanas X* es un sistema gráfico basado en *red* desarrollado por el *MIT* (*Massachusetts Institute of Technology*) en 1984. Se han desarrollado varias versiones de *X*, la más reciente de las cuales es *X versión 11* (**X11**), liberado en 1987. Esta versión se encuentra en su 5a. actualización (release 5).

X11 ha sido adoptado como un estándar de la industria en el manejo de ventanas. Lo soporta un consorcio de líderes de la industria que se han unido para dirigir y contribuir en su continuo desarrollo. Además del desarrollo de *software* dirigido por el *Consorcio X*, muchos desarrolladores independientes producen software de aplicación específicamente para uso con *X*.

La arquitectura del sistema de ventanas está basada en el modelo *cliente-servidor*. El *servidor de X (software)* crea y manipula las ventanas, realiza los gráficos, controla el ratón y el teclado de la *terminal X*. El *cliente* es una aplicación que requiere servicios del *servidor de X*. La comunicación entre ambos se realiza por medio de mensajes en un formato estándar. Este se llama *Protocolo X*, y tiene las siglas *ICCCM* (Inter-Client Communications Conventions Manual). Este protocolo se utiliza aun cuando el *servidor* y el *cliente* se encuentren en la misma máquina. Esto facilita la realización de aplicaciones en *red*. 5.1

5.1 Nabajyoti Barkakati, Xwindow System Programing, p.522

El *protocolo X*, dadas sus características, soporta el procesamiento local (en una máquina), donde el procesador se dedica exclusivamente a las tareas de despliegue, liberando al *servidor* de *aplicaciones* de esta tarea, que consume tiempo, y el procesamiento remoto. El trabajo que queda para el procesador del *servidor* de *aplicaciones* es el de procesar la información de la aplicación, sin ocuparse del despliegue, por lo que la respuesta se vuelve muy rápida.

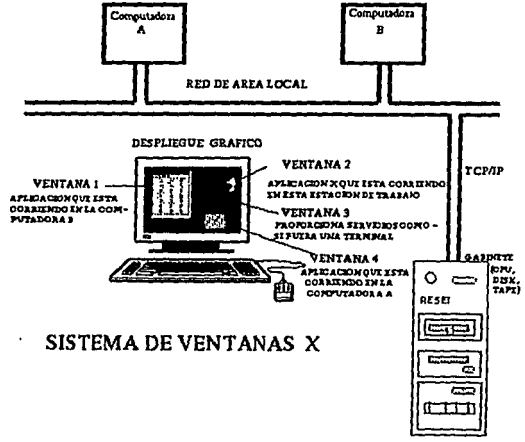


Figura 5.1

En está figura tenemos *servidores de aplicaciones* (computadora A o B), una *terminal X* (*servidor de despliegue*), en *red*; los *clientes* (*ventanas de aplicación*) de diferentes *servidores*, en una *terminal X* ; utilizando el *protocolo X*, el *protocolo TCP/IP* y la administración de servicios de *red*.

5.4 Funcionamiento:

El *sistema de ventanas X* es la interface entre el *protocolo X* y los *clientes*, quien comunica a los *clientes* con su *servidor de aplicaciones* y el *servidor de despliegue (terminal X)*. Todo este intercambio es transparente para el usuario, que finalmente, no se ocupa de cómo están distribuídas las aplicaciones.

Básicamente ofrece un área gráfica de trabajo completamente configurable por el usuario, desde la ventana de fondo de su pantalla, el color del área de trabajo y las preferencias de despliegue, hasta los clientes iniciales que se despliegan por default e interfaces gráficas programadas al gusto del usuario. Por ejemplo: un reloj, una calculadora, etc.

La interface está pensada para facilitar al usuario no experimentado el acceso a sus aplicaciones, y conforme se vaya adaptando al sistema, puede aumentar su complejidad.

La intención de un ambiente gráfico es darle al usuario la oportunidad de aprovechar al máximo la capacidad de procesamiento de sus equipos *servidores*, pues cada ventana es una sesión y no tiene que esperar a que se libere un proceso para iniciar otro, ya sea en el mismo servidor o en algún otro.

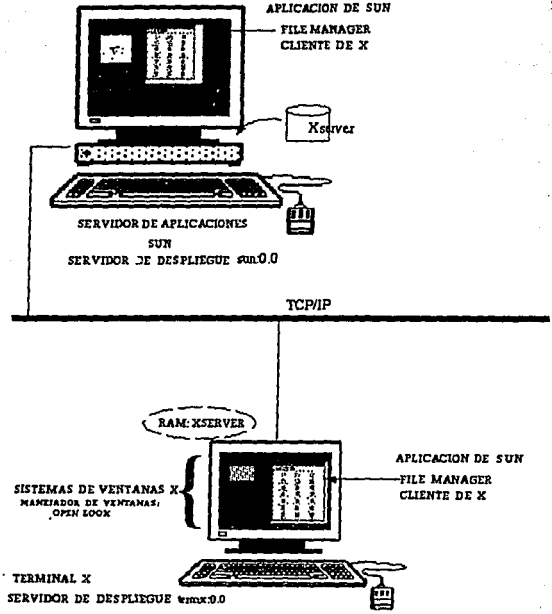


Figura 5.2.

En está figura podemos apreciar el *sistema de ventanas X*, el *servidor de despliegue* y los *clientes*.

5.5 Aplicaciones

Algunos ejemplos de manejadores de ventanas, que utilizan el *protocolo X*:

Motif : Es una interface gráfica basada en *Xlib* cuyas diferencias son casi imperceptibles con el *X Window System* y que da la misma funcionalidad que el *X estándar*.

Open Desktop: Utiliza la misma interface que *Motif*, de modo que no hay cambios perceptibles por el usuario.

Open Look : Es la versión de *Sun Microsystems*, que sin utilizar el juego de aplicaciones de X, soporta todas las versiones de *X11*. Es esencialmente *PostScript (lenguaje de programación y de comunicación gráfica)*, pero al existir divergencias con las versiones de X, se liberó una versión *X11/NeWS (Network Windows System)* en 1987.

6. El sistema operativo *UNIX*

6. El sistema operativo *UNIX*

6.1 Historia

UNIX es un sistema operativo para computadoras desarrollado en los Laboratorios Bell, en Nueva Jersey, Estados Unidos. En 1969, un grupo de investigadores se abocó a la tarea de crear un entorno de programación que facilitara sus labores internas de investigación y desarrollo. Con el apoyo de Dennis Ritchie y de otros investigadores, Ken Thompson, creó un sistema operativo de tiempo compartido, pequeño y de propósito general. Esta primera versión fue escrita en el *lenguaje ensamblador* de una minicomputadora *PDP-7* que ya no usaban; al año siguiente, Ritchie la instaló en una máquina más moderna -una *PDP 11*-, y se dedicó a escribir el compilador para el *lenguaje de programación C*, que acababan de diseñar. En 1973, Thompson y Ritchie escribieron el núcleo de *UNIX* en *lenguaje C;* con ello terminaron la tradición de escribir sistemas operativos en *lenguaje ensamblador* y lograron además, que *UNIX* fuera más portátil y fácil de modificar. Poco después se concedió el permiso para que algunas instituciones no lucrativas tuvieran acceso a *UNIX* en la versión de la *PDP-11*, que ya era muy popular en universidades e instituciones de investigación, y comenzó la rápida difusión del sistema en todo el mundo. En la actualidad, el sistema *UNIX* se considera un estándar virtual para computadoras *multiusuario*, y ha sido adoptado para gran cantidad de máquinas. 6.1

Existen distintas versiones comerciales del sistema :

UNIX III	AIX	Xenix
UNIX V	ULTRIX	etc...
UNIX BSD	HP-UX	
Sun OS	SCO UNIX	

pero todas tienen mucho en común.

6.1 Levin, Revista Ciencia y Desarrollo vol. XIV, núm. 83, pp. 169-178

En 1965, los Laboratorios Bell y la Compañía General participaron en un proyecto de desarrollo de sistemas operativos, integrado al proyecto MAC del Instituto Tecnológico de Massachusetts (MIT), cuyo objetivo era diseñar un gran *sistema multiusuario* denominado *MULTICS*. De esta experiencia se recogieron muchos aspectos que hasta la fecha son importantes en la programación de sistemas, pero el proyecto no culminó. Esto se debió, en parte, a que se trataba de un diseño muy amplio y complejo. Thompson, Ritchie y otros participantes en el proyecto *MULTICS* aprendieron la lección y años después bautizaron su nuevo sistema con el nombre de *UNIX*, el cual tiene una connotación contraria a la de multiplicidad y complejidad.

El nombre *UNIX* proviene de un juego de palabras acorde con la filosofía de su diseño. Cuando Bell Labs abandonó en cierto momento el proyecto *MULTICS*, uno de sus investigadores, Ken Thompson, se lanzó a la búsqueda de algo interesante por hacer. Decidió volver a escribir un nuevo MULTICS, a este nueva versión otro investigador de Bell Labs, llamado Brian Kernighan, lo denominó de manera un tanto burlona *UNICS* (Sistema de información y cómputo con uniplexión). A pesar de los juegos de palabras con "eunucos", para indicar que era un MULTICS castrado, el nombre se afirmó, aunque más adelante se cambió por *UNIX*. 6.2

UNIX y la mayor parte de los sistemas que se ejecutan en él están escritos en *lenguaje C*, y han servido para demostrar que un sistema operativo interactivo no necesariamente es grande y caro, ya sea en equipo o en la cantidad de código, ya que puede utilizarse en *minicomputadoras* de costo reducido, y el desarrollo inicial de su sistema principal requirió menos de 2 años hombre. En palabras de sus creadores, el objetivo es que el sistema sea simple, elegante y fácil de usar.

6.2 Tanenbaum, Sistemas Operativos Modernos, p. 300

Si se recuerda que la función general de un *sistema operativo* es controlar y dirigir la operación de la computadora, de forma tal que presente una imagen monolítica y virtual ante los usuarios del sistema de cómputo, estaremos de acuerdo entonces en que dicho sistema es tan importante como las facilidades físicas y electrónicas de su equipo.

Lo que se espera de un *sistema operativo* es que sea capaz de atender la operación concurrente de múltiples pedidos de atención por parte de los procesos que están ejecutándose en la computadora; que pueda mantener toda la operación bajo control sin perder detalle alguno ni permitir que los procesos interfieran entre sí; que logre un aprovechamiento óptimo de los recursos físicos de la máquina (procesador, memoria, periféricos) y, por ultimo, que haga todo esto en forma silenciosa y eficiente. Como es fácil de comprender, son pocos los *sistemas operativos* que cumplen todos estos requisitos que a veces son incluso contradictorios; no puede esperarse, por ejemplo, que el sistema sea potente, inteligente, eficiente y pequeño al mismo tiempo. 6.3

La razón de la creciente popularidad de *UNIX* reside en que logra combinar facilidad de uso y eficiencia, y en la gran cantidad de ayudas -utilerías- que tiene para programar. Con *UNIX* es sencillo obtener comunicación y sincronización entre procesos, lo que en otros *sistemas operativos* requiere de programación dedicada y exclusiva en los lenguajes de control o incluso, en los más limitados, es virtualmente imposible de lograr.

La filosofía de operación de *UNIX* está basada en el concepto de herramientas de *software*. Esta visión conceptual pide que las tareas computacionales se construyan paulatinamente (de manera que podría llamarse genética); el sistema aporta un conjunto de operaciones primitivas que el diseñador usa para armar aplicaciones que, una vez hechas, pasan a formar parte del acervo de operaciones básicas. Es decir, con un pequeño número de funciones elementales, pueden configurarse programas y sistemas completos que cumplan funciones específicas.

6.3 Levin, op. cit., pp. 169-178

6.2 Características

Entre las características del *sistema operativo UNIX*, se encuentran las siguientes:

Es un *sistema operativo multiusuario* con capacidad de simular *multiprocesamiento* y *procesamiento no interactivo.*

Está escrito en un *lenguaje de alto nivel*: C.

Dispone de un lenguaje de control programable e intérprete de comandos, llamado *shell.*

Ofrece facilidades para la creación de programas, sistemas y el ambiente adecuado para las tareas de diseño de *software.*

Emplea manejo dinámico de memoria (por intercambio o por paginación).

Tiene capacidad de interconexión de procesos.

Permite la comunicación entre procesos.

Emplea un sistema jerárquico de archivos, con facilidades de protección de archivos, cuentas y procesos.

Usa un manejo consistente de archivos de diversos tipos.

Tiene facilidades para *redireccionamiento* de entradas y salidas.

Incluye más de un centenar de subsistemas y varios lenguajes de programación.

Garantiza un alto grado de *portabilidad.* Siempre y cuando se base en alguno de los siguientes estándares: 1003.1 (POSIX), OSF (Open Software Foundation) o UI (UNIX International). Con lo que se logra que si un vendedor de software desarrolla un programa donde sólo

utilice los procedimientos definidos en cualquiera de los tres estándares antes mencionados puede estar seguro de que dicho programa podrá ejecutarse en cualquier sistemas conforme con UNIX. 6.4

El sistema se basa en un núcleo (conocido como *Kernel*), que reside permanentemente en la memoria, y que atiende todas las llamadas del sistema, administra el acceso a los archivos y el inicio o la suspensión de las tareas de los usuarios.
UNIX permite que los programas sean independientes de los dispositivos periféricos; la salida de cada programa o utilería del sistema pueda dirigirse a archivos de disco, impresoras o terminales, y existe también la posibilidad de comunicación entre los procesos para crear conjuntos arbitrarios y complejos de procesos concurrentes cooperativos.

La comunicación con el sistema *UNIX* se da mediante un programa especializado de control llamado *shell.* Este es un lenguaje de control, un intérprete, y un *lenguaje de programación*, cuyas características lo hacen sumamente flexible para las tareas de un centro de cómputo. Debido a esta filosofía, se maneja al sistema con muy pocas órdenes que permiten una gran gama de posibilidades, todo archivo de *UNIX* está controlado por múltiples niveles de seguridad, que especifican los permisos de acceso al mismo. La diferencia que existe entre un archivo de datos, un programa, un manejador de entrada/salida o una instrucción ejecutable se refleja en estos parámetros, de modo que el *sistema operativo* adquiere características de coherencia y elegancia que lo distinguen.

6.4 Tanenbaum, op. cit., p. 304

6.3 Focos de aplicación

UNIX es especial, pues fue diseñado originalmente con una filosofía muy clara y explícita: servir como marco de referencia para desarrollar *software*. Esta marca de origen explica, el gran éxito de *UNIX* entre la comunidad académica y computacional y su relativamente menor penetración y popularidad en el mercado del procesamiento de datos y la informática comercial.

Quienes emplean el sistema *UNIX* como herramienta y entorno de creación de programas y sistemas, encuentran en él un campo extremadamente fértil, hasta podríamos decir exuberante, para sus esfuerzos.

Si bien *UNIX* fue diseñado principalmente con el objetivo anteriormente expuesto, también ha sido explotado en distintas organizaciones comerciales, financieras, científicas, de construcción, etc., ya que las facilidades que ofrece para trabajo en grupos es sumamente versátil y eficiente, ya que permite compartir recursos, ofrece un alto grado de seguridad y una gran gama de *software* de aplicación generado para trabajar en él.

7. El ambiente operativo de Sun

7. El ambiente operativo de Sun

7.1 Clasificación de los diferentes tipos de máquinas dentro de Sun

En Sun Microsystems se tienen varias clasificaciones de máquinas en base a su funcionalidad, mismas que a continuación se describen:

7.1.1 *Server.*

Sistema que provee servicios a las otras máquinas dentro de la red. Un *server* de archivos es una máquina que comparte su almacenamiento de disco y archivos con las otras máquinas que se encuentran dentro de la red. Un *server* es una máquina con un poderoso *CPU*, mayor cantidad de memoria, discos más rápidos y dispositivos periféricos. [7.1]

Un *server* divide el almacenamiento en disco y archivos con otras máquinas dentro de la red. y debe de ser configurado para proporcionar muchos servicios

Es frecuentemente descrito como homogéneo o heterogéneo

Un *server homogéneo* soporta *máquinas cliente* con la misma *arquitectura del kernel.*

Un *server heterogéneo* soporta *máquinas cliente* con diferente *arquitectura del kernel* y con la misma a la vez. Adicionalmente un *server heterogéneo* puede soportar equipo desde otras manufacturas hasta Sun.

7.1 Sun Microsystems, Inc., System Administration 4.1.2, pp.1-2 - 1-7

7.1.2 *Diskless client*

Esta es una *workstation* (estación de trabajo) que no tiene disco propio, para operar enteramente se debe conectar a una red local con *NFS* (Network File System) al *file server*. El *diskless client* se debe conectar al disco de almacenamiento para obtener otros servicios.

Los *diskless client* son generalmente usados por una persona, en la mayoría de los casos también hace las funciones del administrador del sistema.

7.1.3 *Dataless client.*

Es una *workstation* que tiene su propio disco para almacenar operaciones del sistema de archivos. Los *dataless client* usan espacio en disco de un server para almacenar archivos creados por el usuario de la *workstation*.

El disco local debe ser usado para almacenar archivos de datos fuentes. Sin embargo el *dataless client* recibe el directorio */usr* necesario para la operación desde un server vía red.

Esta configuración permite almacenar los archivos ejecutables para muchos *dataless client* en un *server*.

7.1.4 *Standalone.*

Es un sistema completo con un monitor, memoria y disco. El cual no requiere de ningún servicio de red en su secuencia de *boot* (inicialización o arranque).

Puede trabajar en forma independiente a la red.

Los discos contienen el *software* necesario para su operación (una copia del *sistema operativo*).

La mayoría de las máquinas *standalone* pueden cargar *software* y ejecutar respaldos locales

7.2 Ambiente distribuído de Sun

Los sistemas modernos de cómputo son mucho más complicados. Sólo poseen un *CPU* que alcanza niveles de desempeño sólo soñados hasta hace unos pocos años, pero la *red* ha llegado a ser una parte importante e integral de las soluciones de cómputo modernas.

Una *workstation* Sun es usualmente utilizada por un solo usuario, de la misma forma en la que una terminal tiene un usuario. La diferencia radica en que una *workstation* posee su propio *CPU*, memoria y dispositivos de E/S. Esta puede tener discos locales u otros dispositivos periféricos. (Ver figura 7.1).

Una *workstations* Sun puede correr varias versiones del *sistema operativo SunOS* (Sun Operating System). Pueden haber también *workstations* con diferentes *arquitecturas* (sun4, sun4m, sun4c, sun4d, sun4e). Asimismo pueden existir máquinas que no sean *workstations* Sun.

Las características de un sistema de este tipo son:

Un *CPU* (Central Process Unit) compartido por muchas terminales de usuario

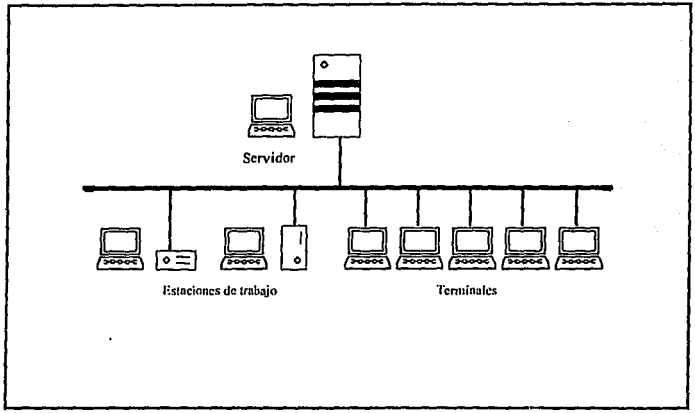


Figura 7.1. Ambiente distribuído de Sun

El almacenamiento en disco es dividido para las terminales que se usen.

La estrategia de respaldo únicamente involucra un sistema y un grupo de discos

Unicamente se permite un *súper usuario*, así la seguridad es razonablemente simple para los privilegios del mismo.

Todos los usuarios del sistema son *usuarios* del mismo nivel de *software* de *UNIX*.

7.3 Ambiente de red SunOS

Estas son las características de un sistema de este tipo:

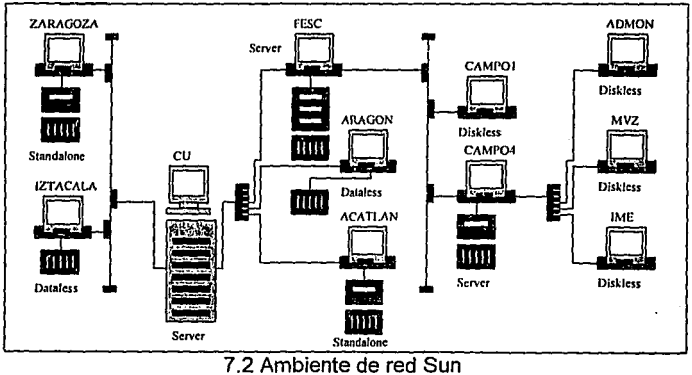
Cada *workstation* SUN tiene su propio *CPU* y memoria

Diferentes categorías de almacenamiento de disco existen para los archivos de las máquinas *server*, *diskless client*, *dataless client* y *standalone*. (Ver figura 7.2)

Las estrategias de trabajo envuelven muchas *workstations* y muchos discos.

Cada *workstation* tiene su propio super usuario, así la seguridad debe ser establecida rápidamente por cada máquina de la red

Sun OS utiliza *NFS* (Network File System)



7.2 Ambiente de red Sun

8. Tareas para la Administración del Sistema Operativo Solaris 2.X

8. Administración del *sistema operativo Solaris 2.X*

8.1. Instalación del *sistema operativo*

Al momento de instalar el *sistema operativo* por primera vez, es recomendable tomar en cuenta los siguientes puntos, para una mayor eficiencia del sistema. [8.1]

1. Planear la instalación antes de efectuarla. Es decir, considerar el tamaño de cada uno de los *file systems*, número de usuarios, paquetes a instalar, manejo de red, etc.

2. Inicializar el equipo a través del *software* del *sistema operativo* ya sea que lo ejecutemos a través de *red*, de *CDROM*, de cinta, etc.

8.1.1 Organización de los file systems.

Generalmente los sistemas asignan valores por default para el espacio que debe de llevar cada *file system* dentro del sistema.

En algunos casos los valores básicos recomendados son:

Servidor

File Systems	Mínimo	Máximo
/	12 *Mbytes*	17 *Mbytes*.
swap	32 *Mbytes*	El default es tres veces el tamaño de la memoria que se tenga en *RAM*.
/usr	30 *Mbytes*	181 *Mbytes*.
/opt	0	Varía según la configuración.

8.1, SilVaTech, Manual de Administración Básica Para Operadores/sss, p. 164

/export/root	0	10 *Mbytes* de base y 20 *Mbytes* por cada máquina *diskless* que se agregue.
/export/swap	0	24 *Mbytes* por cada máquina *diskless* que se agregue.
/export/exec	0	15 *Mbytes* por cada *arquitectura de cliente* (sun4c, sun4m)

Standalone

/	12 *Mbytes*	17 *Mbytes.*
swap	32 *Mbytes*	El default es dos veces el tamaño de la memoria que se tenga en *RAM.*
/usr	30 *Mbytes*	181 *Mbytes.*
/opt	0	Varía según la configuración.

Explicación de las particiones:

1. **root (/)** - Contiene archivos que son únicos para cada máquina, y es la partición sobre la cual van a ser montadas todas las demás particiones, contiene el bloque de arranque y los principales archivos y directorios necesarios para la administración del sistema.

/dev - Primera locación para archivos especiales de dispositivos (por ejemplo, discos y cintas)

/etc - Contiene archivos de administración y bases de datos.

/kernel - Módulos para configuración del *kernel*, binarios y archivos de arranque del sistema.

/sbin - Ejecutables esenciales utilizados en el proceso de inicialización y recuperación manual.

/tmp - Archivos temporales; removidos durante la operación de inicialización.

/var - Archivos de *Log* y otros de tamaño variable como directorios de *spool*.

2. **swap** - Es un área dentro del disco para que se almacene información temporal en el caso de que la memoria del sistema resulte insuficiente.

3. **/export** - Soporte para máquinas de tipo *diskless*.

/export/exec - Ejecutables para máquinas de tipo *diskless*.

/export/root - Particiones para máquinas de tipo *diskless*.

4. **/export/swap** - Archivos de *swap* para máquinas de tipo *diskless*.

5. **/opt** - Generalmente es utilizada para la instalación de paquetes adicionales.

6. **/usr** - Ejecutables y otros archivos dependientes de la *arquitectura*.

/usr/bin - Primera locación para las utilerías estandares del sistema.

/usr/kvm - Binarios y librerías de la arquitectura específica.

/usr/sbin - Ejecutables para la administración del sistema.

7. **/export/home** - Directorios hogar (*home*) de usuarios.

8.1.2 Pasos a seguir para realizar la instalación:

1. Realizar la instalación física del equipo. (Ver apéndice B)

2. Encender el equipo (empezando por los dispositivos externos, monitor y por último *CPU*).

3. Desde el prompt de **OK**, teclear el comando **probe-scsi**, el cual tiene por finalidad detectar los dispositivos de tipo *SCSI* (Small Computer System Interface) que posee nuestra máquina. Si el comando detecta algún error la máquina se quedará congelada, razón por la cual deberemos de apagar el *CPU* y realizar una nueva revisión de las conexiones físicas. Una vez terminado lo anterior, regresar al paso 2.

4. Introducir el sistema operativo e inicializar los servicios de la máquina a través del comando: **b** si es el *prompt* **>** y **boot** si es el *prompt* **OK**; con cualquiera de las siguientes opciones:

ok boot *cdrom* ,

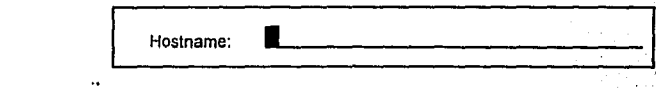
ok boot *tape*,

ok boot *net* , entre otras;

en este caso, será **boot** *cdrom*.

5. Esperar a que se inicialice el sistema y comience el proceso de instalación del *sistema operativo Solaris*.

6. A continuación será inicializado el ambiente gráfico *Open Windows* y será desplegada la siguiente ventana:



En la cual deberá ser introducido el nombre del *host.*

7. La siguiente pantalla que nos será mostrada es:

Esta pantalla tiene por finalidad confirmar si la máquina estará o no conectada en *red.*

8. Posteriormente en el caso de que se hubiera decidido que la computadora estaría conectada en *red*, será preguntada la dirección *Internet* de nuestra máquina. (Ver capítulo 9)

What is your Internet Protocol (IP) address?

The format is four decimal numbers separated by periods (example 129.200.9.1). Use the address that was assigned by local or internet management.

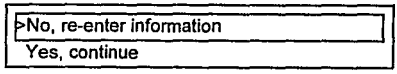
If you have question consult your Networking documentation.

Press Return to continue.

9. La información que se introdujo se confirmará a través de la siguiente ventana:

Is the following information correct?

Hostname: einstein
Connected to network: Yes
IP address: 192.9.200.50

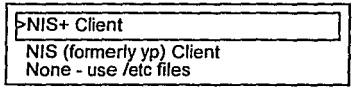


Use the arrow keys to select an item. (CTRL-n next, CTRL-p previous)
Press Return to continue.

Aquí deberá confirmarse o rechazarse la información que se introdujo previamente

10. Una vez que se hubo confirmado los datos anteriores, la siguiente pantalla será desplegada

Do you want to configure this system as a client of a name service? If so, which name service do you want to use? If you do not want to use a name service select 'none' and consult your Install documentation.



Use the arrow keys to select an item. (CTRL-n next, CTRL-p previous)

Press Return to continue.

En la cual se indica que tipo de servicio se requiere o en su defecto que no se desea ninguno.

11. En la siguiente forma de pantalla se nos pregunta si es que nuestra *red* tendrá *subredes*.

Does this workstation's network have sub-networks?

Use the arrow keys to select an item. (CTRL-n next, CTRL-p previous)

Press Return to continue.

En cualquiera de las dos alternativas seleccionarla y pulsar *Enter*.

12. Nuevamente se nos pide una confirmación acerca de la información que se introdujo.

Is the following information correct?

Name service: none
This network is sub-netted: No

```
>No, re-enter information
 Yes, continue
```

Use the arrow keys to select an item. (CTRL-n next, CTRL-p previous)
Press Return to continue.

13. Es necesario indicar la región geográfica en la que nos encontramos

What is your geographic region?

```
 Africa
 Western Asia
 Eastern Asia
 Australia / New Zeland
 Canada
 Europe
>Central America
 South America
 United States
 other - offset from GMT
 other - specify rules file
```

Use the arrow keys to select an item. (CTRL-n next, CTRL-p previous)
Press Return to continue.

14. Luego introducir la zona de tiempo en la cual nos encontramos.

What is your time zone?

```
Mexico / Baja Norte
Mexico / Baja Sur
>Mexico / General
Cuba
none of these - return to regions menu
```

Use the arrow keys to select an item. (CTRL-n next, CTRL-p previous)

Press Return to continue.

15. También deberá ser indicada la hora

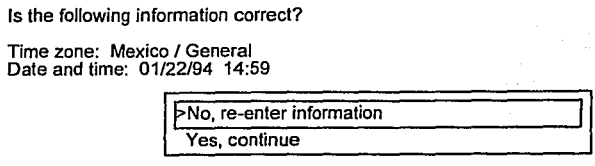
What is the current date and time?
Use digits and fields.

01/22/94 14:59

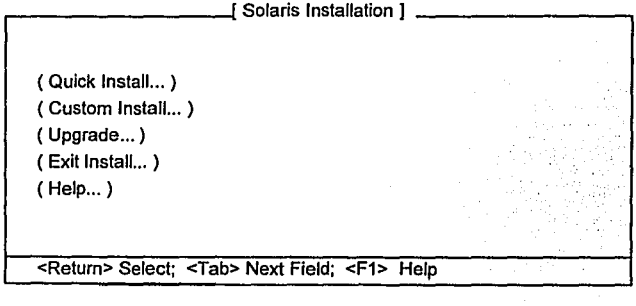
Year	(4 digits)	:	1995
Month	(1-12)	:	07
Day	(1-31)	:	09
Hour	(0-23)	:	17
Minute	(0-59)	:	23

Use the Tab key to move between fields.
Press Return to continue.

16. Es necesario confirmar nuevamente si la información que indicamos es la correcta.

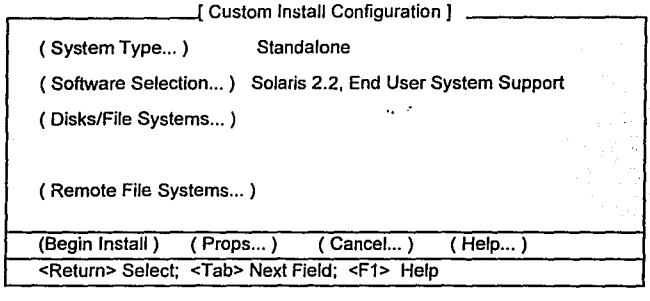


17. A partir de que confirmamos nuestra información será presentada la forma para la instalación de *Solaris*.



En este caso deberá ser seleccionada la opción de *Custom Install.*

18. La forma de *Custom Install Configuration* aparecerá y deberá seleccionarse en primera instancia el campo correspondiente al *System Type*, en el caso de que se desee un tipo de máquina diferente al de *Standalone* que viene por default. Posteriormente se deberá seleccionar el campo de *Software Selection.*



19. Una vez que se haya seleccionado el campo de *Software Selection* habrá que tener en cuenta las siguientes consideraciones acerca de este módulo...

La forma de *software* es usada para seleccionar los programas que serán instalados en el disco local de su *workstation.*

Dentro de la forma de *software* se nos pregunta el tipo de distribución de los paquetes de *software* y grupos. Paquetes, cada uno de los cuales es un grupo instalable de archivos objeto, están clasificados en un grupo que provee para una fácil instalación de diferentes configuraciones de ambientes de *Solaris.*

8.1.3 Paquetes de *software* y grupos

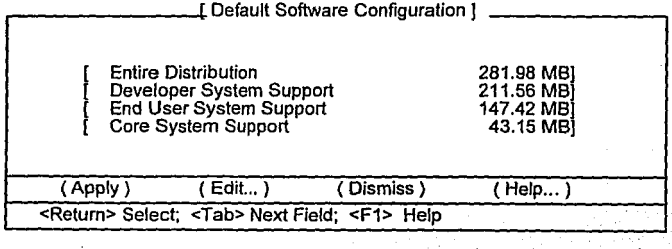
Core - Esta opción contiene solamente el *software* necesario para inicializar el sistema y correr la versión de *Solaris* que se esté ejecutando. Es la configuración mínima de *software*. Si usted instala solamente este grupo, usted no podrá utilizar las utilerías de paquetes, con el objeto de adicionar más paquetes a su sistema.

End User Sistem Support - Esta opción contiene el Soporte del Sistema Core (*Core*) adicional al sistema de soporte para el usuario final como es *Open Windows 3.X*. Esta es la configuración recomendada para el usuario final.

Developer - Esta opción contiene al Sistema de Soporte para el Usuario Final (*End User Sistem Support*) adicional a las librerías y otros archivos de herramientas necesarios para desarrollar *software* en el ambiente de *Solaris 2.X*. Compiladores y depuradores no están incluidos.

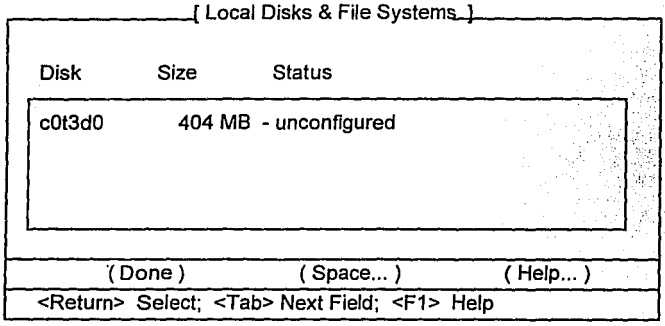
Entire Distribution - Esta opción contiene totalmente la versión completa de *Solaris 2.X*.

Para leer la versión del *software* desde un dispositivo localizado en un *host remoto*, usted tiene que asegurarse que el *hostname* de su máquina esté listado en **/.rhosts** en el *host remoto*.

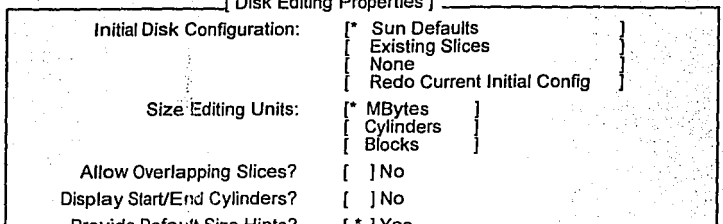


En base a las necesidades del sistema deberá elegirse alguna de las diferentes distribuciones.

20. El siguiente campo que deberá ser consultado es el correspondiente a *Disks/file systems*.



En el caso de que fuera más de un disco se presentará en esta forma, los diferentes discos que se hubieran detectado. Y deberá ser necesario seleccionar a cada uno de ellos y configurarlos individualmente.



La forma de disco es utilizada para introducir la información acerca de el particionamiento de los discos del sistema. Cada disco reconocido por el sistema puede ser configurado con esta forma.

Las categorías en el campo *Initial Disk Configuration* son:

Sun Defaults - utilizar las particiones por default, tamaños, y *puntos de montaje* para este tipo de disco. Usted puede editar cualquier categoría.

Existing Slices - utilizar la *etiqueta del disco* existente. Usted tiene que introducir los *puntos de montaje* a mano. Usted no puede cambiar cualquiera de las particiones o sus dimensiones.

None - En este caso el disco no posee una configuración inicial. Y se podrán introducir los *puntos de montaje a mano*, y usted puede modificar cualquiera de las *particiones* o su tamaño.

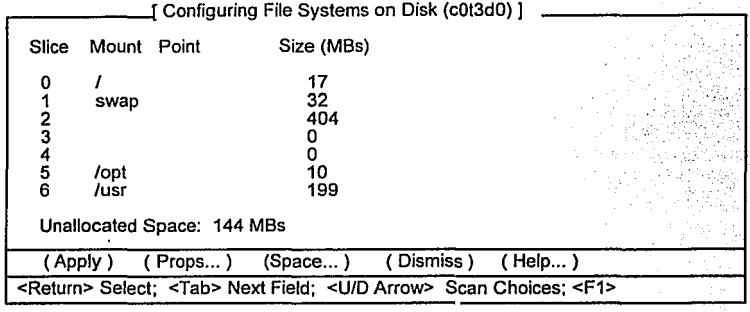
Redo Current Initial Config - Utilizar la *tabla de particiones* salvadas previamente.

En orden para que el sistema de *memoria virtual* trabaje adecuadamente, usted deberá destinar una parte del disco para el área de *swap*.

Cada disco tiene una *partición* designada la *Free Hog Disk Partition.* Esta *partición* se contrae cuando se generan otras particiones o se expande cuando se disminuyen o eliminan otras.

En este caso utilizaremos las opciones que nos da por default el sistema. Y seleccionaremos *Apply*.

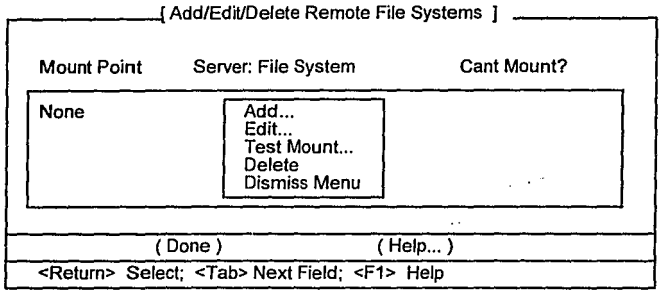
Al haber seleccionado *Apply*, aparece la siguiente pantalla:



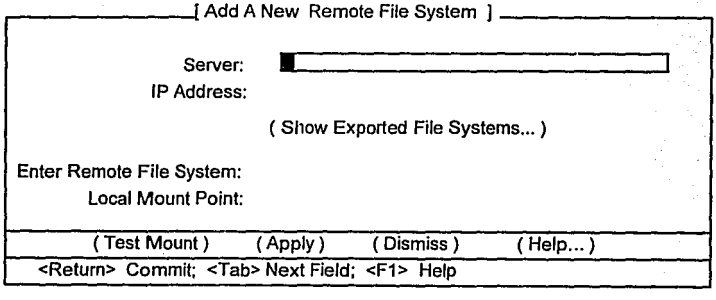
Y en esta se elegirá el tamaño deseado para cada una de las particiones que se deban de generar. Teniendo mucho cuidado de no disminuir el tamaño que trae por default el área de *swap.*

Cuando se haya terminado de usar la forma se deberá seleccionar la opción de *Apply* del menú de *Configuring file systems* y por último de la forma de *Custom Install Configuration* se seleccionará la opción *Done.*

21. Solamente en el caso de existan otras máquinas dentro de la red que estén dando servicios de *NFS* se utilizará la forma correspondiente al campo de *Remote file systems.* Y será desplegada de la siguiente forma.



En el caso de que se seleccione la opción *Add* será desplegada la siguiente forma



Dentro de esta forma se podrán adicionar tantos *file systems* remotos como se deseen. Para salir de esta pantalla, elegir la opción *Dismiss.* Y *Done* dentro de la pantalla principal.

22. Por último, se aplicará la opción de *Begin Install* en la pantalla principal de *Custom Install* y en el caso de que ya no se desee proseguir con la instalación seleccionar la opción *Cancel.*

Finalmente si se desea concluir con la instalación elegir la opción de *Exit Install.*

8.2 Mantenimiento de los file systems en *UNIX*

8.2.1 Estructura de los file systems en *UNIX*

Una partición de un disco puede contener un file system que el *sistema operativo Solaris* interpreta como una jerarquía de directorios y archivos. Cada *partición* puede contener sólo un *file system*, y un *file system* no puede abarcar más de una *partición.*

Un *file system* es creado en una *partición* en el disco. Un *file system* consiste de un número de grupos de *cilindros*, cada uno de los cuales tiene *inodos* (de el término *index node*) y *bloques* de datos. Las estructuras de control de los *file system* son la *etiqueta*, el *superbloque*, los *grupos de bloque de cilindro*, y las *tablas de inodos.*

El *file system* es descrito por el *superbloque*, el cual en turno describe los *grupos de cilindros*. Porque el *superbloque* contiene datos críticos, este es duplicado en cada *grupo de cilindros* para protegerse en contra de cualquier pérdida catastrófica. Esto es realizado cuando el *file system* es creado. Las copias son referenciadas si una falla del disco ocasiona que el *superbloque* sea corrupto.

La *etiqueta* es localizada en el primer *sector* del disco seguido por el *bloque de boot* en los siguientes 15 *sectores*. Juntos, la *etiqueta* y el

bloque de boot hacen un solo *bloque*. Sólo el *file system* de *root* posee una *etiqueta activa* y un *bloque de inicialización*.

El *superbloque* está contenido en los siguientes 16 *sectores*. El *superbloque* contiene información acerca del *file system*, incluyendo el número de *bloques*, el número de *grupos de cilindros*, el tamaño de un *bloque* y *fragmento*, las preferencias de optimización, una descripción del *hardware*, el nombre del *punto de montaje*, y otros campos.

Cuando un *file system* es *montado*, una copia del *superbloque* es hecha en memoria. Todas las subsecuentes operaciones para crear o remover archivos implica modificaciones en la memoria del mismo. Una causa común de corrupción en el *file system* es una ruptura del *sistema operativo*. Debido a que el sistema está tratando de igualar la información de la memoria a la información del *superbloque* del disco.

Al comienzo de cada *grupo de cilindros* se localiza el *superbloque de respaldo*.

El *file system* es estructurado de manera que la información del *grupo de cilindros* es almacenada en un ramal flotante desde el inicio del grupo de cilindros.

El *bloque del grupo de cilindros* describe el número de *inodos* y el número de *bloques de datos*, el número de directorios, el número de *bloques* libres, *inodos* libres y los *fragmentos* libres en el *grupo de cilindros*, la lista de *bloques* libres, y el mapa de *inodos* usados.

8.2.2 Conceptos relacionados con File Systems

/etc/vfstab Este archivo lista todos los *file systems* que tiene *montados* el sistema.

Directorio de montaje Directorio utilizado para *montar* un *file system*. Es decir, es el directorio con el cual será relacionada una *partición* del disco, directorio remoto o diskette.

mount Montar un *file system* hacia el directorio al que tendrá acceso el usuario.

mountall Montar todos los *file systems* contenidos en el archivo **/etc/vfstab**

umount Desmontar un *file system*.

umountall Comando para desmontar todos los *file systems* contenidos en el archivo **/etc/vfstab**.

inicialización Generar un *file system* sobre un disco lógico usando el comando **mkfs**.

fsck pass Número asociado con un *file system*, especifica el orden en el cual el **fsck** checa los *file systems* al momento del *boot*.

mount options Permisos de acceso al *file system*

/etc/dfs/dfstab Archivo que contiene todos los *file systems* que pueden ser exportados.

share Comando para poner a disponibilidad los *file systems* contenidos en el archivo **/etc/dfs/dfstab**

shareall Comando para exportar múltiples *file systems*.

dfshares Comando para verificar los recursos que están disponibles

nfs.server Comando para inicializar los demonios de *NFS*

nfs.client Comando para inicializar los demonios del *automounter*

unshare Comando para imposibilitar el acceso a los *file systems* contenidos en el archivo **/etc/vfstab**

unshareall Comando para denegar permisos de acceso a los *file systems* proporcionados a través de *NFS*.

8.2.3 Operaciones sobre los File Systems:

Crear *file systems*.

Agregar *file systems*.

Borrar *file systems*.

Modificar *file systems*.

Montar o *desmontar file systems*.

Efectuar *respaldos* de un *file system* completo.

Recuperar información de un *respaldo*.

Las operaciones sobre un *file system* se pueden efectuar por medio de un menú, si existe esa opción, o bien directamente en el archivo **/etc/vfstab**.

El formato del archivo **/etc/vfstab** es el siguiente:

% **cat** */etc/vfstab*

```
#device    device    mount     FS      fsck    mount     mount
#to mount  to fsck   point     type    pass    at boot   options
#

venus:/    -         /venus    nfs     -       yes       -
```

Los campos son:

device to mount	El nombre del *servidor* separado por dos puntos (:) y la ruta del recurso remoto.
device to fsck	Los recursos *NFS* nunca son checados por los clientes, por eso en muchas ocasiones este campo permanece nulo.
mount point	El punto de *montaje* por default para el *file system.*
FS type	Siempre **nfs** para los recursos de *NFS.*
fsck pass	Los recursos *NFS* nunca son checados por los *clientes*, por eso en muchas ocasiones este campo permanece nulo.
mount at boot	**yes** o **no** indican que el *file system* puede ser *montado* cuando el sistema entra al nivel 2, o cuando el comando **mountall** es utilizado.
mount options	La lista de opciones del comando **mount**, separadas por una coma.

El archivo **/etc/dfs/dfstab** contiene las particiones a ser exportadas por los comandos **share** y **shareall**. El formato del archivo **/etc/dfs/dfstab** es el siguiente:

```
$ cat /etc/dfs/dfstab
# Introducir aquí los comando share para realizar una ejecución
#automática al entrar o ejecutar al nivel 3.
#
# share [-F fstype] [-o options] [-d "<text>"] <pathname> [resource]
# .e.g,
# share -F nfs -o rw=engineering  -d "home dirs" /export/home2
share -F nfs -o ro /usr/share/man
```

8.2.3.1 El comando mount

El comando **mount** es utilizado para agregar un *file system* local o remoto.

Formato del comando:

mount *[-F nfs] [-o options] server:pathname mount-point*
Si el *file system* se encuentra disponible dentro del archivo **/etc/vfstab**, usted puede especificar tan sólo el *server:pathname* o bien el *mount-point*, ya que el comando **mount** siempre consulta el archivo **/etc/vfstab** para mayor información.

 # mount */usr/share/man*

Si el *server:pathname* y el *mount-point* son especificados, y la opción *-F* es omitida, el comando **mount** toma el tipo del *file system* del archivo **/etc/vfstab**.

Opciones:

-F **nfs** Especifica **nfs** como el tipo de *file system*. Esta opción no es requerida ya que **nfs** es el tipo por default para un *file system* remoto.

-o **options** Especifica separando a través de comas, la lista de opciones específicas tales como **rw** para montar el recurso como *read-write*, **ro** como *read-only* (el default es **rw**)

server:pathname Especifica el nombre del servidor separado por dos puntos (:) y la ruta de acceso del *file system* deseado.

mount-point Especificar la ruta de acceso del *punto de montaje* en el sistema local (el cual debe de existir).

8.2.3.2 El comando umount

El comando **umount** es utilizado para detallar que *file system* (local o remoto) será desmontado.

Formato del comando:

umount *[-F nfs] server:pathname o mount-point*

La línea del comando puede especificar tanto el *server:pathname* o el *mount-point.*

 # umount */usr/share/man*
#

Opciones:

-F **nfs** Especifica **nfs** como el tipo de *file system*. Esta opción no es requerida ya que **nfs** es el tipo por default para un *file system* remoto.

8.2.3.3 El comando share

El comando **share** pone en disponibilidad los recursos para ser *montados* por sistemas remotos. Si ningún argumento es especificado, después **share** despliega todos los recursos actualmente compartidos.

Formato del comando:

share *[-F nfs] [-o options] [-d description] pathname*

Opciones:

-F **nfs** Esta opción puede ser utilizada con el comando **share** dentro del archivo **/etc/dfs/dfstab** para el script **shareall** a fin de trabajar correctamente. Esta opción no es requerida en la linea de comando porque **nfs** es el default para el tipo de *file systems* remotos (que se encuentra listado en el archivo **/etc/dfs/fstypes**)

-o **options** Especifica una lista separada por comas de las opciones posibles para los *file systems*. Esas opciones incluyen **rw** para asignar permisos de *read-write* para todos los *clientes*, **ro** para limitar el acceso a *read-only* para todos los *clientes*, **rw**=*client[:client]...* para permitir acceso de lectura y escritura para lo *clientes* listados, **ro**=*client[:client]...* para permitir acceso solamente de lectura a los *clientes* listados, y *root=host[:host]...* para permitir a un *cliente* huésped tener acceso de *root*.

-d **description** Provee un comentario que describe los *file systems* que serán compartidos. Este comentario es desplegado por el comando **share** sin argumentos.

pathname Especifica el nombre de la ruta de acceso del *file system* que será compartido.

8.2.3.4 El comando unshare

El comando **unshare** hace que los *file systems* no estén disponibles para la realización de *montajes* remotos.

Formato del comando:

unshare *[-F nfs] pathname*

Opciones:

-F **nfs**	Especifica **nfs** como el tipo de *file system*. Esta opción no es requerida ya que **nfs** es el tipo por default para un *file system* remoto.
pathname	Especifica el nombre de la ruta de acceso para el *file system* que será deshabilitado para ser compartido.

8.2.3.5 Comunes mensajes de error de *NFS* y posibles soluciones

La mayoría de los problemas son descubiertos a través de los mensajes de consola o los síntomas en un cliente.

Mensaje de error:

nfs mount: mars:: RPC: Name to address translation failed - n2a: hostname not found

Este mensaje puede aparecer durante el *boot* o en respuesta a una petición explícita de *montaje* e indica un *servidor* desconocido.

Solución:

Checar que el nombre del *host* se encuentre en la tabla de **hosts.**

Mensaje de error:

NFS mars not responding, still trying

Este mensaje puede aparecer durante el *boot* o en respuesta a una petición explícita de *montaje* e indica que un *servidor* conocido es inaccesible.

Solución:

1. Checar que el *servidor* no esté caído o fuera de servicio.

2. Checar para ver si la *red* entre la *estación de trabajo* y el *servidor* no está fuera de servicio a través del comando **ping.** [8.2]

Mensaje de error:

nfs mount: mars:: RPC: Program not registered

Este mensaje puede aparecer durante el *boot* o en respuesta a una petición explícita de *montaje* e indica que un *servidor* conocido es inaccesible porque no están corriendo en él uno o más demonios de *NFS.*

Solución:

1. Utilizar **who** *-r* en el *servidor* para checar si éste se encuentra en el nivel 3. Si no es así, cambiarse al nivel 3 con el comando **init 3.**

2. Usar **ps -e** en el *servidor* para checar que demonios de *montaje* y de *NFS* están corriendo. Si no se encuentran, iniciarlos con el *script* **/etc/init.d/nfs.server** y la palabra clave **start.**

Mensaje de error:

nfs mount: mars:/opr: No such file or directory

Este mensaje puede aparecer durante el *boot* o en respuesta a una petición explícita de *montaje* e indica que un nombre de *file system* desconocido dentro del *servidor.*

Solución:

Checar que directorios existen dentro del *servidor* y escribirlo correctamente en el archivo **/etc/vfstab.**

8.2 Sun Education Services, System Administration II, pp. 2-17 - 2-21

Mensaje de error:

mount: mount-point /DS9 does not exist.

Este mensaje puede aparecer durante el *boot* o en respuesta a una petición explícita de *montaje* e indica que el *punto de montaje* no existe.

Solución:

Checar que el *punto de montaje* exista en el *cliente* y esté escrito correctamente en la línea de comando en el archivo **/etc/vfstab.**

Mensaje de error:

le0: No carrier - transceiver cable problem?

Este mensaje puede aparecer durante el *boot* o en respuesta a una petición explícita de *montaje* e indica problemas de red.

Solución:

Checar las conexiones físicas de la *red* entre su máquina y el *servidor* (incluyendo los terminadores).

Mensaje de error:

stale NFS file handle

Este mensaje puede aparecer cuando un proceso intenta accesar un archivo remoto y el archivo indicado está fuera de fecha.

Solución:

El archivo puede haber sido movido del *servidor. Desmontar* y *montar* nuevamente el recurso en el *cliente.* Usted puede tener este mensaje ***"nfs mount: masr:/usr/share/man: No such file or directory."*** En ese caso, contactar con el administrador del *servidor* y y preguntarle acerca de los *file systems* perdidos.

8.2.4 Puntos a considerar para agregar, borrar y modificar file systems.

- Especificar el directorio en el cual se va a *montar.*

- El nombre de la partición del disco o del *file system* deseado.

- Los *file systems* remotos requieren:

- Nombre del *host*

- Directorio remoto para *montarlo.*

- Permisos de lectura y escritura (*rw*).

- Exportar el *file system* si es remoto.

8.2.5 Observaciones para montar y desmontar file systems

- Especificar el directorio sobre el cual se va a *montar* el *file system.*

- El *montaje* se hace sobre *file systems* disponibles.

- Se *desmonta* un *file system* disponible.

- Para efectuar un *montaje* se utiliza el comando **mount**.

- Para *desmontar* un *file system* se utiliza el comando **umount**.

8.2.6 El File System de ROOT (/)

Es el principal file system, de él se desprenden todos los demás y contiene entre otras cosas:

- El mínimo de comandos requeridos.

- Los archivos de inicialización del sistema (**ufsboot**).

- Directorios a partir de los cuales se efectuará el *montaje* de otros *file systems*.

- El *file system* de *root* contiene archivos específicos de la máquina y directorios cruciales para la operación del sistema.

- Los directorios importantes son:

dev El directorio de dispositivos. Contiene archivos de dispositivos y el *script*, **MAKEDEV**, para crearlos a ellos.

etc El directorio de administración. Archivos importantes como **passwd, hosts, ethers**, y los *scripts* **rc.**

sbin El directorio de archivos binarios de una estación de trabajo configurada como *standalone*. Esos son los archivos que necesita el sistema para permitir que comience el *sistema operativo*: **init**, **mount**, y **sh**.

tmp Archivos temporales creados por el sistema de ventanas o el *compilador de C* están localizados aquí. Todos los archivos son borrados de este directorio cada vez que el sistema es *reinicializado*.

var Los archivos de las cuentas y de *spool* están localizadas y/o son creadas en este directorio. Algunos administradores localizan el directorio de **/var** en su propia *partición*.

8.2.7 El file system /usr

Los directorios dentro del file system de **/usr** contiene comandos ejecutables, programas de sistema, y librerías de rutinas. Los directorios importantes son:

5bin Contiene programas específicos para el ambiente del *Sistema V*. Hay además un **5include**, el cual contiene archivos de cabeceras del *Sistema V*, y **5lib**, el cual contiene librerías del *Sistema V*.

bin El directorio de binarios del sistema. Este es donde la mayoría de los programas de usuario, como **date**, **ls**, y **pwd** están localizados.

kvm El *software* que especifica la arquitectura del *kernel* está localizado en este directorio.

lib Las librerías del sistema están localizadas aquí, a lo largo con utilerías misceláneas, macro paquetes y otro *software*.

share Contiene archivos de texto y datos que pueden ser compartidos a través de diferentes *arquitecturas* de sistema. Los manuales en línea, los directorios **man** están localizados aquí.

ucb Este directorio tiene los programas de la Universidad de California, Berkeley. Este incluye **dbx**, **rlogin**, y **vi**.

8.2.9 El file system /export

El *file system* de **/export** contiene directorios que el servidor hace disponibles para la *red*. Algunos de esos archivos son exportados a una restringida lista de *clientes*, típicamente los directorios de *root* y de *swap*.

root Los directorios de *root* de los *clientes*. Un directorio jerarquizado es creado para cada *cliente* y contiene un *kernel* y otros sistemas de archivos específicos.

share Contiene el *software* que es independiente de la *arquitectura*. Usualmente está implementado como una liga simbólica a **/usr/share**.

exec Este directorio contiene los archivos de **/usr**. Un *cliente monta* **/usr**, y **/export/exec/kvm/arch.rel** como si fuera propio **/usr/kvm** (**arch.rel** significa **arch**itecture.**rel**ease). Cuando un *cliente* tiene la misma *arquitectura* del *kernel* que el *servidor* al cual está conectado, puede compartir los directorios **/usr** y **/usr/kvm** del *servidor*. De otra manera, los archivos específicos para la *arquitectura* del *cliente* serán cargados dentro de el directorio **/export/exec** del *servidor*.

8.2.9 Otros file systems

El *file system* **/export/swap** está donde los *swapfiles* están localizados. Los *swapfiles* son grandes archivos de espacio de disco prelocalizado que un *cliente* utiliza como una memoria secundaria. El contenido de esos archivos es temporal y por ello indeseable para respaldar. Un *swapfile* es creado para cada *cliente* del *servidor*.

El file system **/home** es donde se sitúan los directorios *home* de los usuarios. Por convención, el *path* para los directorios de *home* es **/home/hostname/loginname**. Frecuentemente, los directorios de

home están localizados en el *servidor* de archivos y *montados* en **/home** para el resto de la *red*.

9. TCP/IP

9. TCP/IP

9.1 Capas de la arquitectura de TCP/IP

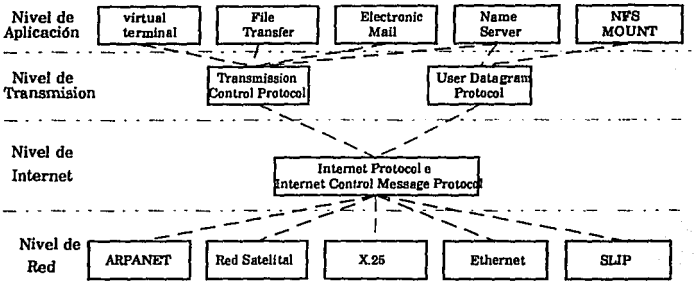


Figura 9.1

TCP/IP se puede implantar en casi cualquier medio físico de comunicaciones de datos y protocolos de las capas física y de enlace de datos .

Como se ve en la figura 9.1 se muestran los protocolos que se encuentran en cada capa de *TCP/IP*, los que pertenecen a las tres primeras capas (*Aplicación, Transmisión e Internet*) son los que se explicaran en los temas siguientes, el *nivel de red* solo ejemplifica sobre que protocolos de este nivel es posible establecer una comunicación vía *TCP/IP*.

9.2 Protocolos

En la figura 9.2 se muestran los protocolos que intervienen en la comunicación con el protocolo *TCP/IP* comparados con el modelo *OSI*.

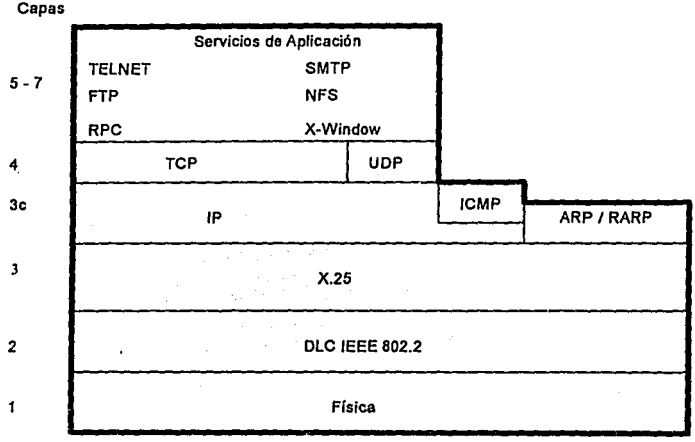


Figura 9.2

9.2.1 ARP (Address Resolution Protocol)

El host emisor utiliza el Address Resolution Protocol (Protocolo de Resolución de Direcciones), *ARP* para obtener la dirección *Ethernet* del host destino.

ARP es el proceso que construye la liga de dirección entre el nivel Internet y el nivel de Interfase de Red. Es usado por un host para preparar una unidad de información para transmisión en red.

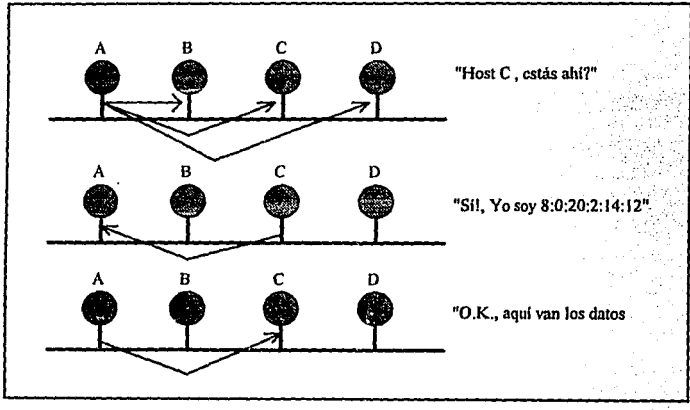


Figura 9.3

Existe una tabla *ARP* que se mantiene en memoria y almacena las requisiciones más frecuentes de direcciones *Ethernet*. La tabla *ARP* es consultada cada vez que se requiere una dirección *Ethernet* para un destino, para preparar el frame *Ethernet* para transmisión (Ver figura 9.3).

Si una dirección *Ethernet* no aparece en la tabla *ARP*, se resuelve en una dirección *IP* por el host emisor. Este hace la requisición de la dirección *Ethernet* en la red enviando un broadcast tipo *ARP* a todos los hosts en la red local. El host con la dirección *IP* correspondiente contesta con su dirección *Ethernet*.

9.2.2 RARP (Reverse Address Resolution Protocol)

El Reverse Address Resolution Protocol (Protocolo de Resolución de Direcciones Inverso) (*RARP*) es una variante de el Protocolo de Resolución de Direcciones (*ARP*). *RARP* solamente traduce direcciones, pero en dirección opuesta, es decir convierte direcciones *Ethernet* en direcciones *IP*, en lugar de direcciones *IP* a direcciones *Ethernet*.

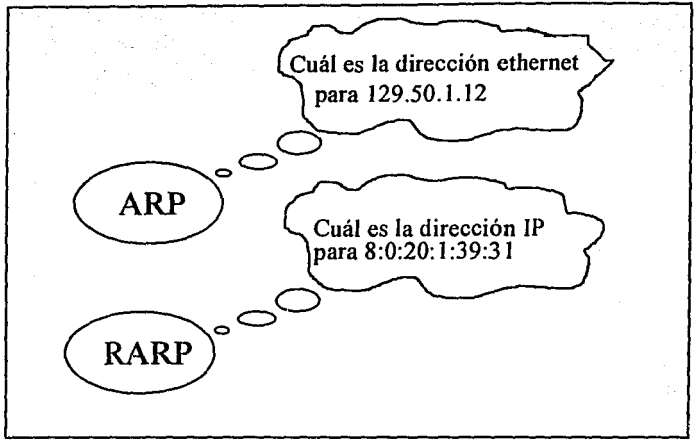


Figura 9.4

RARP Ayuda a configurar los sistemas *diskless* y permite a las *workstationss diskless* el leer su dirección *IP* (Ver figura 9.4). Un sistema *diskless* no posee disco que le permita leer la configuración de *TCP/IP*, es decir no almacena su dirección *IP*. Sin embargo cada sistema conoce su dirección *Ethernet* por que esta codificada en la Interfase *Ethernet* del hardware (Tarjeta de comunicación o motherboard si la tiene integrada).

Cuando un servidor de la red observa un requerimiento, busca la dirección *Ethernet* en la tabla de /etc/ethers, contesta con la dirección

IP de la workstation.

Ejemplo de archivo /etc/ethers.

```
gerardo% more /etc/ethers
08:00:20:05:38:79       svt
00:00:a7:11:99:e5       tarjeta
00:00:a7:12:11:c7       caput
00:00:a7:11:cc:fc       sol
00:80:a3:04:4b:6c       EPS_044B6C
00:00:a7:11:70:e1       godzilla
00:00:a7:12:20:6d       ncd5
00:00:a7:12:20:be       contabilidad
00:00:a7:11:cf:78       Winston
00:00:a7:12:bf:ac       admsilvia
00:00:a7:13:10:fe       desastres
00:00:a7:12:56:05       mcx2
00:00:a7:11:5b:4c       atila
00:00:a7:11:cc:d2       israel
00:60:8c:ba:bf:96       Tieta
00:40:10:50:1b:96       mac
00:00::a7:11:cc:d2      otello
00:00:a7:10:d8:45       idea
00:00:a7:12:20:e7       oso
00:00:a7:11:cd:3b       ncd1    #15b de prueba
00:00:a7:12:1f:5d       ncd2
00:00:a7:11:cc:f9       ncd3
```

El archivo /etc/ethers es un archivo simple de texto que se puede crear usando un editor.

Es necesario este archivo solamente si su sistema provee servicios a *workstations diskless*; de otra manera puedes no tenerlo.

Se puede hacer el archivo /etc/ethers teniendo en líneas sencillas la dirección *Ethernet* separado de la dirección el nombre del host separado por tabuladores.

Los nombres de los hosts son asignados por el administrador de la red,

pero la dirección *Ethernet* es asignada por el fabricante, y por lo tanto se obtiene de la interfase de la red.

La mayoría de las *workstations* UNIX despliegan su dirección *Ethernet* mientras estan Booteando.

9.2.3 IP (Internet Protocol)

Formato de el *datagrama* de Internet Protocol (*IP*) con la cabecera *IP* (Ver figura 9.5).

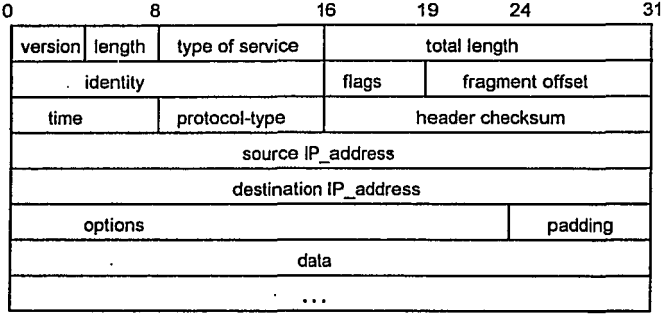


Figura 9.5

Un elemento importante almacenado en el encabezado *IP* es la dirección anfitrión nodo destino; la unidad resultante es un *datagrama IP*.

En general un *datagrama* puede definirse como un paquete de longitud finita con información suficiente para ser enviado en forma

lograr la comunicación:

netid: Es el identificador de la red Internet.

hostid: Es el identificador del host de la red.

IP_Address => netid.hostid

El número de bytes asignados a cada parte(*netid* y *hostid*) varia, dependiendo de la clase de red.

En la practica cada dirección *IP* debe pertenecer a un cierto tipo de red que a continuación se describen:

Para su determinación del tipo de red al cual se hace referencia, se dan los siguientes criterios:

Las redes comprendidas entre los parámetros de 1.X.X.X a 126.X.X.X son redes de clase A.

Las redes comprendidas entre los parámetros de 128.X.X.X a 192.X.X.X son redes de clase B.

Las redes comprendidas entre los parámetros de 193.X.X.X a 223.X.X.X son redes de clase C.

La dirección 127.X.X.X no es usada puesto que se utiliza para realizar pruebas internas de reconocimiento (loopback).

Por la razón de que una dirección Internet usa ambos parámetros de identificación de red como de host,(*netid* y *hostid*), cada vez que se refiere a una dirección de Internet, se refiere a una conexión de una red y no a una máquina en particular.

El caso es de que un gateway conectado n redes con n distintas

direcciones *IP*, posee n conexiones a través de las redes diferentes.

La dirección Internet puede ser utilizada para referirse a redes en particular así como a una máquina en particular. Para determinar una red en particular la dirección del *hostid* será todos los bits en cero.

Por ejemplo:

132.43.0.0 = 10000100.00101011.00000000.00000000

En donde 132.43 es el *netid* y 0.0 es el *hostid.*

Lo que indica que se esta refiriendo a una red de tipo B con dirección de red 132.43 como identificador de red e identificadores de hosts los 16 bits restantes.

Es decir la principal característica del Protocolo *IP* es la identificación de nodos.

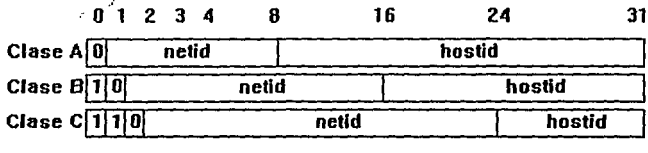


Figura 9.6

Las tres clases de direcciones Internet (*IP* address); de las cuales las tres primeras A,B y C se determinan por los primeros dos bits (Ver figura 9.6).

Clase A

10.0.0.32(10) = 00001010.00000000.00000000.00010000(2)

Clase B

128.14.58.60(10) = 10000000.00001110.00111010.00111100(2)

Clase C

192.9.150.202(10)= 11000000.00001001.10010110.11001010(2)

El software de Internet esta diseñado alrededor de tres servicios de red arreglados de manera jerárquica, la característica del formato de la arquitectura de *TCP/IP* es lo que lo hace robusto y muy adaptable. Conceptualmente la red Internet provee tres tipos de servicios diferentes, donde se definen como se muestra en la siguiente figura 9.7.

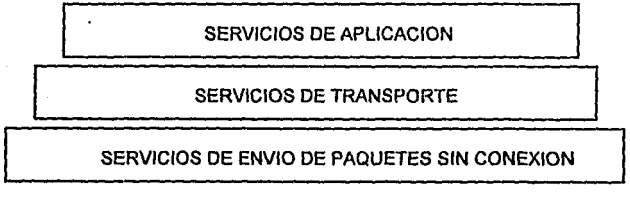


Figura 9.7

En el nivel mas bajo se encuentra el servicio de envío de paquetes sin conexión, que es el ofrece los servicios a las capas superiores. En el

siguiente nivel se encuentra la capa de transporte que sirve de plataforma para el desarrollo de servicios de aplicación.

9.2.3.1.1 Direccionando y clasificando una red

Para ejecutar correctamente TCP/IP en una red es necesario que se tenga un número IP, y cada host tenga una dirección IP. La dirección de una máquina consiste de la siguiente información.

net number	*subnet number*	*host number*
Asignado por NIC	Asignados por el administrador	

De esta forma se determina el número de dirección IP a usar:

- **net number**

Es posible el obtener este número a través del *NIC* (Network Information Center), es posible el asignar este número pero se corre el riesgo de que si se tiene conexión vía la red Internet se encuentre duplicado el *net number* provocando un conflicto.

- **subnet number**

Este es un número que es asignado por el administrador de la red y que debe ser único para cada subred que se desee controlar.

- **host number**

Este es un número único que identifica alguna máquina dentro de la red y que es asignado por el administrador de la red.

9.2.3.2 Dirección Ethernet

Es la dirección física, la cual se define por hardware. Mediante la cual cada sistema identifica sus nodos dentro de una LAN, (Ver figura 9.8 y 9.9)

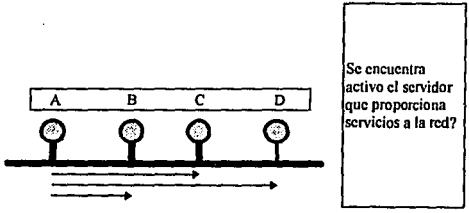


Figura 9.8

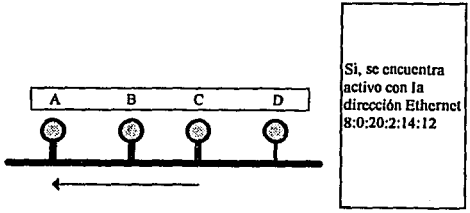


Figura 9.9

9.2.4 TCP (Transmission Control Protocol)

El protocolo de control de transmisión (*TCP* Transmission Control Protocol) es un protocolo de la capa de transporte y de esta manera descansa sobre el protocolo *IP*.

Su principal tarea es la transportar de manera confiable datos a través de la red.

Es conveniente para grandes volúmenes de información y para envío de información que debe viajar a través de *ruteadores* y *gateways*.

Los principales atributos de *TCP* son:

- Proporciona conexión *full-duplex*

Las conexiones *TCP* proveen de transferencia concurrente en ambas direcciones. Desde el punto de vista de la aplicación, una conexión *full-duplex* consiste de dos flujos independientes de información en direcciones opuestas. El software del protocolo puede regresar información de control para un flujo en el *datagrama* que a su vez trae información en la dirección opuesta. Esto reduce tráfico en la red.

- Conexión Virtual del Circuito

Antes de que la transmisión pueda iniciarse, las aplicaciones tanto emisora como receptora deben interactuar con sus sistemas operativos. Esta interacción informa al sistema operativo para preparar lo que sea necesario para que se pueda iniciar la comunicación. Esto

es análogo a hacer una llamada telefónica. La línea debe estar establecida antes de poder empezar a hablar.

- Asegura la transmisión de datos usando:
  - Números de secuencia
  - Checksum
  - Acuse de recibo
  - Retransmisión de un segmento después de un time-out de acuse de recibo
  - Direccionamiento de puerto utilizando un número de puerto de 16 bits

La unidad de datos que son usadas por *TCP* son llamadas *segmentos* y se constituyen de la siguiente forma: (Ver figura 9.10)

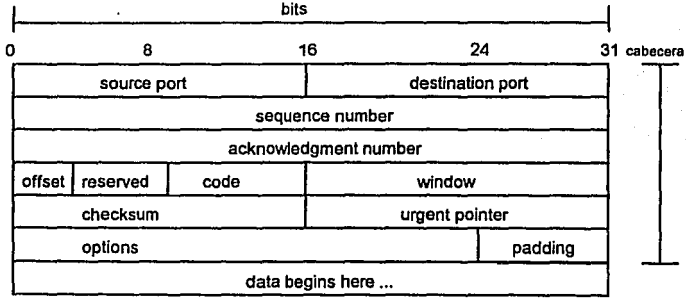


Figura 9.10

Cada segmento contiene una suma de seguridad (*checksum*) que se envia a quien recibe los datos y que verifica que los datos se recibieron intactos.
Si el segmento de datos es recibido dañado, el que envia recibe un acuse de recibo positivo (*acknowledgment*) y vuelve a enviar el segmento. Después de un tiempo apropiado el que envia retransmite cualquier segmento hasta que no recibe un acuse de recibo positivo.

TCP es una conexión orientada (*connection-oriented*), es decir establece de manera lógica una conexión punto-a-punto (*end-to-end*), entre dos hosts.
El control de la información es llamado handshake y es el intercambio entre dos puntos finales que establecen un dialogo después de que los datos son transmitidos.
TCP indica la función de control en un segmento en el campo de la cabecera en un bit en el campo Flags.

El tipo de handshake usado por *TCP* se llama three-way handshake por que son tres segmentos los que son intercambiados.

En la figura que se presenta se ve gráficamente el manejo de los handshake (Ver figura 9.11)

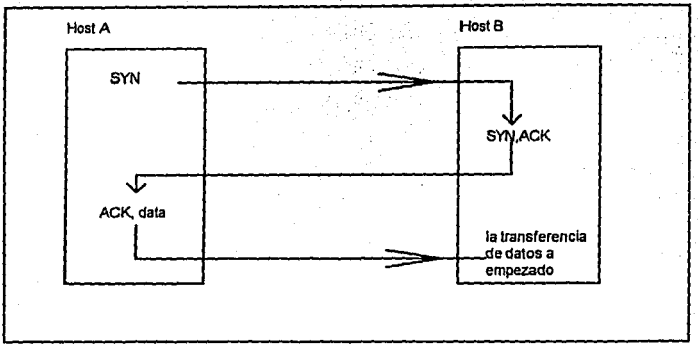


Figura 9.11

El Host A empieza la conexión enviando al Host B un segmento con la secuencia de números de sincronía (Syncronize sequence numbers - SYN), este segmento dice a B que A desea establecer una conexión, y B le contesta que número de secuencia de A es el que va a usar para esos segmentos.

El host B responde a A con un segmento que es el acuse de recibo ACK y el bit SYN. En el host B el segmento de acuse de recibo recibe el segmento de A y le informa a A cual es el número de secuencia del host B en el cual va a empezar. Finalmente el host A envia un segmento de acuse de recibo al segmento de B y transfiere el primer

dato.

Después de este intercambio el *TCP* del host A ha recibido evidencias positivas de que el host remoto esta listo para recibir datos. Y como la conexión esta estable los datos se empiezan a transferir.

Cuando los módulos han sido terminados de enviarse realiza un handshake con un segmento que contiene un bit de "no mas datos por enviar" (este bit es llamado FIN) y se cierra la conexión.
Esta manera de cambiar datos punto-a-punto provee una conexión lógica entre dos sistemas.

TCP recibe los datos que entran de la aplicación y la divide en paquetes llamados segmentos. Los segmentos se componen por un encabezado, como el que usa *UDP*, pero este contiene mas información de control así como los números de puertos seguidos por la sección de datos.

9.2.5 UDP (User Datagram Protocol)

UDP es un protocolo sin conexión, sin estado. Esta diseñado para pequeñas transmisiones de datos que no requieren un mecanismo de transporte confiable.
Esta aplicación permite el intercambio de mensajes sobre la red con

un mínimo de la cabecera del protocolo (Ver figura 9.12).

Formato de la cabecera UDP

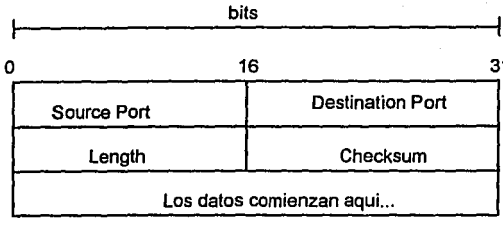


Figura 9.12

UDP no es un protocolo seguro, establece una conexión tipo conectionless, como se ve en la cabecera, éste protocolo no contiene técnicas que permitan verificar que los datos han sido enviados del otro lado de la red correctamente.

Dentro de la computadora, *UDP* determina la entrega de datos correctamente. *UDP* usa 16 bits para el número puerto fuente y el puerto destino, en la palabra 1 de la cabecera.

Los principales atributos de *UDP* son:

- Establece una conexión tipo conectionless
- Direccionamiento vía número de puertos
- checksums
- muy sencillo
- no espera respuesta de acuse de recibo

9.2.6 ICMP (Internet Message Control Protocol)

Los errores en la comunicación ocurren todo el tiempo en todas las redes y en todos los nodos.

La manera en la cual se notifica que un error está ocurriendo es a través del protocolo *ICMP* (*Internet Control Message Protocol*).

ICMP es una parte componente de la implementación *IP* y su papel es como un protocolo de transporte en el cual sus únicas tareas son las de diagnosticar y transportar el error para el Internet Protocol.

La estructura que se muestra en el siguiente diagrama es común para todos los mensajes de *ICMP* y contiene los siguientes campos:
(Ver figura 9.13)

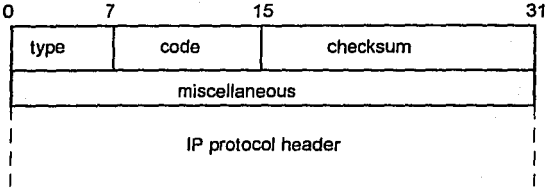


Figura 9.13

Type

Este campo especifica el tipo de mensaje de *ICMP*

Tipo Campo	Funcion
0	Echo reply
3	Destination unreachable
4	Source quench
5	Redirect
8	Echo request
11	Time exceeded for a datagram
12	Parameter problem on a datagram
13	Time stamp request
14	Time stamp reply

Figura 9.14

En el diagrama anterior (Ver figura 9.14) las aplicaciones más importantes son:

source quench: Es un aviso al emisor de detener temporalmente la transmisión de datagramas.

redirect messages: Un gateway puede avisar al host de utilizar otra ruta.

echo request / reply: Mensaje con el fin de saber si un sistema remoto se encuentra activo.

Code

Es el código mas lejano que indica las subfunciones con un tipo.

Ckecksum

Este campo contiene una suma internet para mensaje completo de ICMP

Miscelaneous

Este es un campo de 32 bits que contiene información de propósitos miscelaneos (número de secuencia, Internet addresses, etc.)

IP protocol header

Este campo contiene el datagrama enviado y los primeros 8 bits del mensaje transportado.

9.3 Servicios de Aplicación

9.3.1 TELNET

Una sesión *telnet* se puede definir como una terminal virtual de red.

El comando *telnet* es usado en la máquina cliente, mientras en el servidor se ejecuta un demonio conocido como telnetd (Ver figura 15).

El servicio de ***telnet*** es usado por el protocolo *TCP* por el puerto 23.

Esta aplicación y el proceso *telnetd* son una conexión que es llamada pseudo terminal, la aplicación de este programa nos permite tener una interfase como en una terminal normal.

Diagrama esquematico de una sesion TELNET bajo UNIX

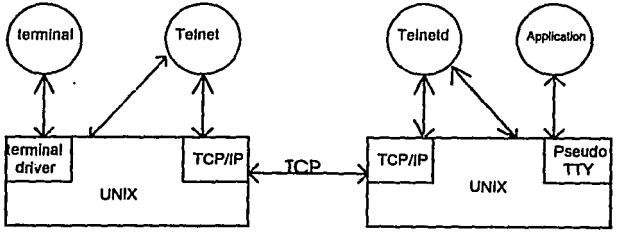


Figura 9.15

El siguiente es un ejemplo del uso del comando telnet. La conexión será realizada al nodo terror, en el cual existe un usuario llamado terror.

 # telnet *terror*

Trying 192.9.200.60 ...
Connected to terror.
Escape character is '^]'.
UNIX(r) System V Release 4.0 (sparc20)

login: terror
Password:
Last login: Tue Oct 11 16:21:15 from terror
Sun Microsystems Inc. SunOS 5.3 Generic September 1993
$
.
.
.
Cualquier instrucción o aplicación (basada en desplegado ASCII) es posible usarla
.
.
$ exit
Connection closed by foreign host.

9.3.2 File Transfer Protocol (FTP)

A diferencia de *TELNET; FTP* establece un canal de comunicación con la posibilidad de transferir archivos.

Dentro del archivo /etc/services utiliza el puerto 21.

El comando del protocolo tiene el mismo nombre *ftp* en el servidor se llama *ftpd.*

FTP difiere en muchos aspectos de otros programas de transferencia de archivos.

La diferencia mas prominente es que usa es que incluye el uso de canales separados para el control de la información y los datos, la transferencia de archivos no puede ser ejecutada en *background.*

FTP usa texto *ASCII.*

Para cada transferencia de datos se establece una conexión *TCP* entre el cliente y el servidor, la misma que es cerrada después de la transmisión.

De esta forma *FTP* usa rasgos de *TCP* y usa todas las precacuciones para tener una transmisión libre de errores.

Existen dos formas de transmisión, los dos modos son implementados en sistemas UNIX, nombradas modo texto y modo binario; en el modo texto los archivos son enviados como líneas *ASCII* separadas por retornos de carro y símbolos de nueva línea.

En el modo binario, un archivo es transmitido sin conversión, como una secuencia de bytes, naturalmente es mejor y mas rápido.

9.3.2.1 Comando ftp

Para empezar con una sesión *FTP*, es necesario establecer una sesión *ftp* en el servidor. Para ejecutar una transferencia de archivos es necesario obtener acceso al archivo. Es necesario tener un logging en el server, asi como un *password* puesto que *ftp* lo toma como parte del protocolo para establecer la comunicación.

Los siguientes comandos son usados frecuentemente.

Comandos FTP	**Función**
bell	Realiza un sonido al termino de cada transferencia.
get	Extrae un archivo del servidor.
mget remote-files	La función de *get* pero con múltiples archivos.
rename from to	Renombra el archivo remoto "de" "hacia".
put	Envia un archivo al servidor.

mput local-files	Función de colocar múltiples archivos locales en la máquina remota.
put local-file[remote-file]	Envía un archivo local con el nombre remoto especificado, en caso de omitir el mismo se enviara con el nombre local.
del	Borra un archivo en el servidor.
delete remote-file	Borra un archivo en la máquina remota.
binary	Establece un modo de transmisión binario.
ascii	Establece un modo de transmisión *ASCII*.
cd	Cambia de directorio en el servidor.
lcd	Cambia de directorio en el cliente.

cdup	Cambia del directorio actual hacia un directorio arriba.
pwd	Muestra el *path* en el servidor.
dir	Muestra el contenido del directorio en el server.
hash	Muestra "#" para cada bloque transmitido.
quit	Finaliza el comando *ftp*.
bye	Da por terminada la sesión *FTP*.
quit	Sinónimo de *bye*.
close	Termina la sesión de *FTP* con el servidor remoto eliminando los macros existentes.
disconnect	Sinónimo de *close*.
help	Imprime información del comando especificado.

open host[port] Establece una comunicación con el servidor de *ftp*, puede especificarse un puerto en particular; generalmente el servidor se encuentra en *auto-login* (default), por lo que *ftp* siempre intenta abrir este.

user user-name [password] [account]
Identifica el servidor remoto como un usuario en particular dentro del mismo.

? [command] Sinónimo de *help*.

Esquemáticamente *FTP* trabaja de la siguiente forma: (Ver figura 9.16)

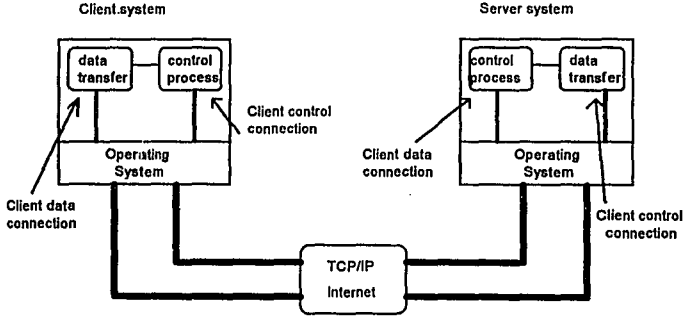


Figura 9.16

El siguiente es un ejemplo usando ftp. En el primer ejemplo, la conexión ftp estará hecha con el nodo terror. El directorio local será /classic y el directorio remoto será /pypsa/informix/israel/finales. Los archivos .4gl serán copiados del directorio remoto al local.

 # ftp terror

Connected to terror.
220 sparc20 FTP server (UNIX(r) System V Release 4.0) ready.
Name (terror:root): root
331 Password required for root.
Password:
230 User root logged in.
ftp> pwd
257 "/" is current directory.
ftp> cd /pypsa/informix/israel/finales
250 CWD command successful.
ftp> lpwd
/classic
ftp> status
Connected to terror.
No proxy connection.
Mode: stream; Type: ascii; Form: non-print; Structure: file
Verbose: on; Bell: off; Prompting: on; Globbing: on
Store unique: off; Receive unique: off
Case: off; CR stripping: on

Ntrans: off

Nmap: off

Hash mark printing: off; Use of PORT cmds: on

ftp> prompt

Interactive mode off.

*ftp> mget *4gl*

200 PORT command successful.

150 ASCII data connection for b_ascii.4gl (192.9.200.100,32838) (5819 bytes).

226 ASCII Transfer complete.

local: b_ascii.4gl remote: b_ascii.4gl

5994 bytes received in 0.023 seconds (2.6e+02 Kbytes/s)

ftp> dir

200 PORT command successful.

150 ASCII data connection for /bin/ls (192.9.200.100,32908) (0 bytes).

total 31562

-rw-r--r-- 1 israel informix 0 Oct 18 11:28 .cdd

-rw-r--r-- 1 israel informix 0 Oct 18 11:28 .cts

-rw-r--r-- 1 israel informix 0 Oct 17 08:54 a.cdd

-rw-r--r-- 1 israel informix 0 Oct 17 08:54 a.cts

-rwxr-xr-x 1 israel informix 24576 Nov 2 08:22 aleat

5 ASCII Transfer complete.

12142 bytes received in 0.57 seconds (21 Kbytes/s)

ftp> !ls

ao compara.4gl giros.4gl proto.4gl reporcen.4gl

azp control.4gl lost+found proto2.4gl reporcuad.4gl

ftp> bye

221 Goodbye.

9.3.3 Trivial File Transfer Protocol (TFTP)

El *TFTP* es un protocolo de transferencia de archivos con un mínimo de requerimientos.
Dentro del archivo /etc/services usa el puerto 69.
Como en *FTP, TFTP* solamente soporta la transmisión en modo binario y texto.
La principal diferencia con *FTP* es que usa una conexión tipo connectionless, *UDP* en nuestra caso.
Los códigos del protocolo *TFTP* son:

- *Read Request (RRQ)*
- *Write Request (WRQ)*
- *Send Data (DATA)*
- *Acknowledgement (ACK)*
- *Error (ERROR)*

La comunicación entre el servidor *TFTP* y el cliente empieza con los paquetes *RRQ* y *WRQ*. De esta manera, los *ID's* son intercambiados para que mas tarde sean identificados los paquetes de datos. Ellos solo contienen el nombre de los archivos y el modo de la transmisión que será usada. Los paquetes *DATA* son usados para la transmisión de datos actuales y el servidor recibe un *ACK* o un *ERROR* del paquete. La transmisión es cerrada cuando un paquete *DATA* menor a 512 bytes

es recibido.

9.3.3.1 Comando tftp

Como es usual, en los sistemas UNIX existe un comando *tftp* correspondiente al protocolo *TFTP*. En el servidor el proceso es llamado *tftpd*. Las opciones disponibles para *tftp* son muy similares a las del comando ftp, pero con mas restricciones (por ejemplo los procesos de autorización).

Diagrama de funcionamiento de TFTP: (Ver figura 9.17)

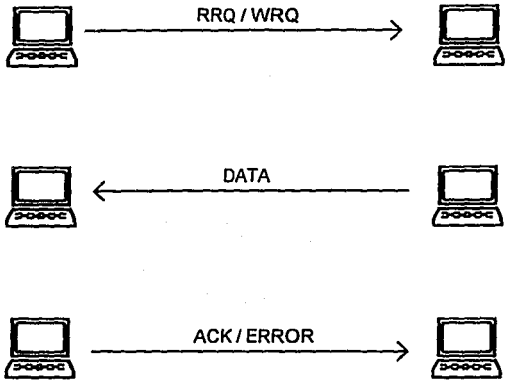


Figura 9.17

El siguiente es un ejemplo del comando tfpt, el servidor al cual nos conectaremos es terror, usando al usuario israel, y copiaremos el archivo principal.4gl del directorio /pypsa/informix/israel/finales, al directorio local /pruebas.

tftp terror

tftp> ?

Commands may be abbreviated. Commands are:

connect	*connect to remote tftp*
mode	*set file transfer mode*
put	*send file*
get	*receive file*
quit	*exit tftp*
verbose	*toggle verbose mode*
trace	*toggle packet tracing*
status	*show current status*
binary	*set mode to octet*
ascii	*set mode to netascii*
rexmt	*set per-packet retransmission timeout*
timeout	*set total retransmission timeout*
?	*print help information*

tftp> connect

(to) 192.9.200.60

tftp> status

Connected to 192.9.200.60.

Mode: netascii Verbose: off Tracing: off

Rexmt-interval: 5 seconds, Max-timeout: 25 seconds

tftp> ascii

tftp> mode

Using netascii mode to transfer files.

tftp> get /pypsa/informix/israel/finales/principal.4gl principal.4gl
tftp> quit
#

9.3.4 Simple Mail Transfer Protocol (SMTP)

Es un protocolo Internet para el correo electrónico, el puerto 25 de *TCP* es definido para *SMTP*, al igual que *TELNET* y *FTP*, *SMTP* se distingue por su simplicidad e incorpora muchos de los rasgos de *FTP*.

sendmail

En sistemas UNIX, *SMTP* es implementado por el programa /usr/lib/sendmail. Este es un programa muy complejo en el cual es posible el comunicarse con otros servicios de correo (*SMTP*).

9.4 Comandos Remotos

9.4.1 rsh (Remote Shell)

El remote *shell rsh* y la correspondencia en el servidor *rshd* (usa a TCP en el puerto 514) permite al usuario el ejecutar comandos en una computadora diferente. La línea de comandos es como un parámetro que se envia al servidor, una conexión *rsh* al sistema UNIX tiene una

entrada y salida estándar en los canales *stdin, stdout* y *stderr* son comandos que se inician cuando el proceso corre por medio de dos conexiones *TCP*. De esta manera muchos de los comandos UNIX pueden ser usados sin alteración.

rsh tiene algunas restricciones:

- El código de salida del comando en el servidor no es regresado.
- En los procedimientos de shell, no es posible checar si el comando fue exitosamente ejecutado.
- Los programas terminal-oriented, así como los editores no pueden correr bajo un *rsh* o corren con restricciones.

9.4.2 rlogin (Remote login)

El comando *rlogin* permite tener un *login* en una computadora diferente, el proceso en el servidor es llamado *rlogind*.

Como en una sesión TELNET, un *rlogin* inicia un procedimiento en el servidor y una pseudo terminal.

Funcionalmente tiene muy pocas diferencias con telnet en un sistema UNIX.

El protocolo *rlogin* es muy simple, al comenzar la sesión tres cadenas de caracteres separadas por ceros, son enviadas: ('[1]...[2]')

[1] Esto contiene el nombre original (*login*) en la computadora cliente, llamado bajo el ID cuando rlogin es iniciado.

[2] Este contiene el nombre *login* en el servidor, (TERMINAR)

[3] Este contiene el nombre *login* y la transmisión (*rate*) de la terminal y que es usado para trabajar en la máquina cliente (por ejemplo, vt100/9600). El *ID* de la terminal es transferido al servidor a la variable *TERM*, y las aplicaciones con comenzadas, enviándose apropiadamente el control de caracteres en la terminal. La velocidad de la conexión es almacenada como información de la estructura interna de la pseudo terminal.

rlogin puede no usar otro protocolo; todos los caracteres tecleados en el teclado son transparentemente enviados al servidor y todos los caracteres regresados son desplegados en la pantalla del usuario en la computadora cliente.
rlogin es terminado desde el shell local, por ejemplo tecleando ^D. Si el carácter de control no tiene efecto, una terminación explícita es el introducir "~" al comienzo de la línea.

9.4.3 rcp (Remote Copy)

El comando *rcp* es utilizado para transferir archivos y opera como una extensión del comando cp en la red.

Todas las aplicaciones de *cp* pueden ser usadas en la red (por ejemplo copiado de archivos locales a una computadora remota y viceversa, copiado entre tres computadoras y copiado local) son implementadas por *rcp*.

Posibles aplicaciones del comando *rcp*:
Donde ***terror, frankie*** y ***lobo*** son servidores y ***sam*** es un usuario del servidor ***frankie*** y ***joop*** es un usuario de lobo.

- Comando normal
  $ rcp terror:/etc/passwd frankie
  $ rcp terror frankie: .
  $ rcp terror: frankie /etc/passwd ? sam

- Transferencia entre tres computadoras
  $ lobo:/etc/passwd terror: /etc/passwd

- Usando otro usuario ID
  $ rcp sam lobo@joop $HOME

9.5 Procesos para el buen funcionamiento de la Red.

Para el correcto funcionamiento de una red con TCP/IP deben de existir los siguientes procesos:

ifinit/dev/loop0 tcip

Debe de existir un proceso ifinit por cada tarjeta ethernet

Otros procesos requeridos son:

/usr/bin/smtp
/usr/bin/rwhod
/usr/bin/inetd

9.6 Chequeo de la instalación de TCP/IP

Al instalar el TCP/IP debe de tomarse en cuenta algunos detalles de importancia, que son:

El asignar el nombre y dirección del host en el cual se instala, así como el de los que se encuentran en la red.

En algunos casos debe decidirse la dirección Internet de brodcast.

Todos 1's si es compatible con BSD 4.3
Todos 0's si es compatible con BSD 4.2.

Además del nombre del dispositivo de salida, el cual depende del equipo en el que se está trabajando.

9.7 Instalación y modificación de redes.

Para la instalación y modificación de los parámetros tanto del *TCP/IP* como de *NFS* existen en la mayoría de los sistemas la posibilidad de realizar estos movimientos por medio de un menú.

Sin embargo, todas las modificaciones se pueden efectuar directamente en los archivos correspondientes. Es decir, la dirección y el nombre de un nuevo host se puede agregar en el archivo */etc/host* utilizando un editor de texto.

***%cat** /etc/hosts*
#
#Sun Host Database
#
#If the NIS is running, the file is only consulted when booting
#

```
127.0.0.1 localhost
#
192.9.200.2     Atila
192.9.200.3     Cali
192.9.200.4     Napoleon
192.9.200.5     Morgan loghost
192.9.200.6     Tieta
192.9.200.8     Mafalda
192.9.200.9     Adolf
192.9.200.10    Xaviera
192.9.200.7     galileo
192.9.200.11    Elena
192.9.200.12    Winston
192.9.200.13    Kris_O.
#192.9.200.14   ncdserv1
192.9.200.15    Godzilla
192.9.200.16    Brismark
#192.200.17     Tandem
192.9.200.21    ncd1
192.9.200.22    ncd2
192.9.200.23    ncd3
192.9.200.24    ncd4
192.9.200.25    ncd5
192.9.200.26    ncd6
```

192.9.200.27 ***ncd7***
192.9.200.50 ***comte***
192.9.200.14 ***dlink***
192.9.200.42 ***pcxware***

9.8 Direcciones Ethernet

Las direcciones Ethernet son utilizadas por *TCP/IP* (Transmission Control Protocol and Internet Protocol) y *NFS* (Network File System) para identificar a los hosts.

Las direcciones se dan de alta en el archivo **/etc/ethers**, pudiendo hacerse directamente sobre ese archivo.

La dirección *Ethernet* está compuesta por dos dígitos en forma hexadecimal, está compuesta por 6 números separados por dos puntos (:).

Ejemplo:

08:00:12:la:11:lb

El archivo ethers contiene la dirección ethernet del nodo de la red.

%cat /etc/ethers

#

00:80:3f:23:00:7e	Mamushka
00:00:a7:00:79:71	MCX
00:00:a7:10:d8:45	Napoleon
08:80:3f:23:00:28	Paulina
02:60:8c:a:3:2c:fe	Tieta
02:60:8c:a7:bb:0b	Xaviera
02:60:8c:6c:31:3b	Mafalda
00:40:c8:00:0f:14	xerox
02:60:8c:6c:31:3a	CLIENTE1
00:00:A7:11:5b:4	Atila
08:00:20:03:93:c4	Morgan
00:00:a7:11:cc:f9	Adolf
00:00:a7:11:cf:78	Winston
00:00:a7:11:cc:d2	Kris_0.
00:00:a7:11:70:e1	Godzilla
00:00:a7:12:05:e3	Brismark
00:00:a7:12:05:d7	ncd1
00:00:a7:11:99:e5	ncd2
00:00:a7:12:20:76	ncd3
00:00:a7:12:20:7d	ncd4
00:00:a7:12:20:6d	ncd5
00:00:a7:12:1f:5d	ncd6

00:00:a7:12:20:c8 ***ncd7***
08:00:20:03:fd:21 ***comte***
00:80:c8:0b:6e:06 ***dlink***
00:40:c8:02:07:9a ***Elena***
02:60:8c:a3:2c:fe ***pcxware***
%

La primera columna indica la dirección *ethernet* y la segunda el nombre de la máquina.

9.9 Parámetros de TCP/IP

Dentro de la configuración y definición de *TCP/IP* es necesario definir los siguientes parámetros:

- Nombre del Host
- Dispositivo a usar dentro del directorio **/dev** (inen0,loop,le0,e0,etc)
- Dirección Internet

9.10 NFS (Network File System)

NFS permite que directorios y archivos sean compartidos a traves de la red. Este sistema fue desarrollado originalmente por SUN Microsystems, pero ahora es soportado por casi todas las implementacionesa UNIX y muchos otros sistemas no UNIX.

A traves de NFS, usuarios y programas pueden accesar archivos localizados en un sistema remoto, como si fueran archivos locales.

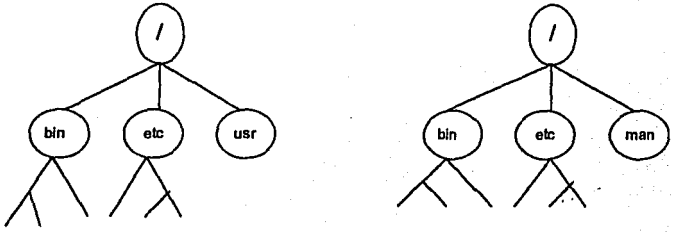
NFS tiene diversos beneficios.

- Reduce el almacenamiento local por que una red puede almacenar una copia de un directorio, mientras que el directorio continua siendo accesado por todos los miembros de la red.

- NFS centraliza las tareas centralizando las aplicaciones.

- NFS permite a los usuarios no familiarizados con UNIX el tener acceso a recursos de la red sin necesidad de usar comandos de transaccion de archivos como *ftp, o rcp* solo es necesario usar el comando cp.

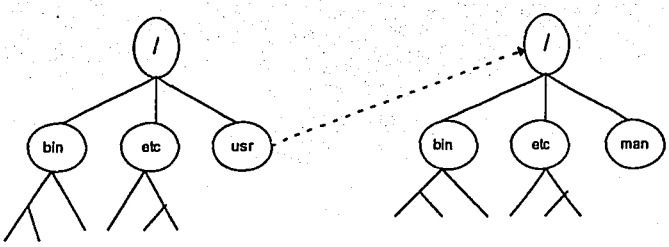
El siguiente es un ejemplo de dos sistemas distintos los cuales en el sistema A posee /usr y el sistema B necesita correr ciertas aplicaciones las cuales se encuentran en /usr de la máquina A. (Ver figura 9.18).



Sistema A

Sistema B

Figura 9.18



Sistema A Sistema B

Figura 9.19

Cuando se comparte un file system de un disco local, se comparte todo lo que contenga este, cuando tu compartes una particion, en la máquina local se va a poder accesar como un directorio de la máquina remota (Ver Fig. 19).

10.1 Exportando servicios via NFS.

En un servidor, el programa exportfs el demon rpc.mount y el demon nfsd controlan NFS. Los file systems o los archivos que se van a exportar son dados de alta en */etc/exports*, cuando se bootea un servidor nfs el archivo de booteo /etc/rc.local automaticamente activa el programa exportfs, este programa ve los archivos que se encuentran en */etc/exports* e informa al kernel acerca de los permisos para cada uno de los servicios que seran exportados.

Es posible el exportar servicios manualmente con el comando *exportfs* , sin embargo todos los servicios deben de estar contenidos en el archivo exportfs.

Terminologia usada para definir NFS

Servidor de NFS	Sistema que provee los recursos de los file systems a la red.
Cliente de NFS	Sistema que usa los file systems a través de la red.
Archivo remoto	Archivo que reside en otro sistema diferente al que está conectado el usuario.
Archivo local	Archivo que está en el sistema al que está conectado el usuario

Punto de montaje Nombre del directorio en el que un file system es montado y hecho disponible a los usuarios.

9.10.2 Accesando servicos de NFS

Un cliente va a accesar los archivos de el servidor **montando** los archivos exportados por el servidor de NFS.
El proceso de montaje usa una serie de llamadas que hacen disponer al cliente de un directorio o archivo de manera transparente como si fuera un directorio de el servidor.
La manera en la cual se recibe informacion de el servidor es a traves de un protocolo llamado RPC (Remote Procedure Calls), en un formato especial este formato es llamado XDR (EXternal Data Representation).
La informacion es procesada por el demonio rpc.mountd, y el cliente permite (o no) el montar el archivo jerarquicamente.

9.10.3 Mantenimiento de un servidor NFS

Es necesario el realizar las siguientes actividades para tener un rendimiento correcto del servicio NFS.

- Adiciona las entradas necesarias en el archivo */etc/exports*
- Cambia los permisos si es necesario en */etc/exports*, asegura que los usuarios puedan leer y solamente root pueda escribir.
- Durante una sesion de trabajo es posible el exportar y desexportar directorios o archivos.
- Si se tiene en la red mas de un servidor NFS en la red, es posible que se haya decidido el que uno de ellos sea cliente de otro servidor de NFS, esta no es una estrategia recomendada, es posible que no se pueda montar jerarquicamente un directorio de otro servidor, sin embargo si se asume el papel de cleinte es posible el compartir estos recursos sin dañar el performance de el servidor para otros posibles clientes.

9.10.4 Demonios NFS

NFS se ejecuta con diversos demonios, algunos le dan performance (rendimiento) a las funciones del cliente y algunas le dan performance a las funciones del server.

Los demonios que son usados por NFS son los siguientes:

nfsd	El demon nfsd, corre en el servidor de NFS, este demonio sirve a los requerimientos del cliente.
biod	Este demonio corre en el cliente y se encarga de las entradas y salidas del cliente.
rpc.lockd	Este demonio se encarga de tomar los requerimientos del cliente y del servidor.
rpc.statd	Este demonio se encarga de monitorear la red, es tambien requerido por rpc.lockd para proveer servicios de monitoreo de red. En particular este demonio permite el cerrar correctamente despues de una ruptura. Este corre en el cliente y en el servidor.
rpc.mountd	Este demonio es el encargado de los procesos de montaje del cliente, este demonio corre en el servidor de NFS.

10. Terminales X

10.Terminales X

10.1 Descripción

Una *terminal X*, es un *servidor de despliegue*. Variando con cada fabricante, éstas terminales cuentan con memoria propia, opcionalmente el *servidor de X* residiendo localmente; tienen un procesador que se dedica a realizar las tareas de despliegue y con ésto liberan al procesador del *servidor* de la carga.

El trabajo que queda para el procesador de cada máquina (*host*) con el que la *terminal X* establezca sesiones es el de procesar la información de la aplicación que se trabaje, y no su despliegue.

Por su gran flexibilidad, se integran a cualquier sistema existente, y permiten tener aplicaciones corriendo en diferentes *hosts* desplegándose en una sola terminal. Su administración se centraliza en el *servidor de X*, sin que esto limite sus posibilidades de conexión a otros *hosts*.

El ciclo de vida de una *terminal X* no es tan limitado como el de cualquier otro *hardware* que se evalúe. La actualización de una *terminal X* depende únicamente del *software*, es decir, del *servidor de X*.

Una *terminal X* virtualmente ofrece capacidad ilimitada de *multitarea* que las terminales *ASCII* no pueden dar, así como la capacidad de manejar aplicaciones gráficas.

Hay que tener muy claro que, aún cuando las *terminales X* (y su *software* del servidor X) ofrecen la capacidad de utilizar un ambiente gráfico, no está determinando qué manejador de ventanas se va a utilizar, sino que puede ser cualquiera que el usuario prefiera. La *terminal X* da al usuario el servidor de despliegue propicio para sacar provecho de sus conexiones sin tener que invertir mucho en *hardware*, o tener que subir la capacidad del sistema (upgrades/actualizaciones). Es decir, se puede utilizar *protocolo X* y *sistema de ventanas X* para comunicar una *PC* con un *servidor UNIX*, sin embargo, todo el proceso lo hará el *servidor*. Las *terminales X* toman parte de esta tarea, utilizando su memoria y su procesador.

10.2 Arquitectura Cliente-Servidor

El uso de éstos términos en X puede parecer en un principio diferente de su uso en cualquier otro contexto computacional. Para X, el *servidor de X* es el *software* que maneja un despliegue, teclado y mouse. El *cliente* es el programa desplegado en el monitor que toma entradas del teclado y mouse.

El *cliente* puede estar corriendo en la misma máquina que el *servidor* o en otra máquina en la *red*.

Servidor de X. El *servidor*, o *Xserver*, es una pieza de *software* que corre en una máquina determinada para ello, no necesariamente el *servidor* de aplicaciones, pero sí con los servicios de *red* suficientes.

Este *software* establece la comunicación entre el *cliente* y su *servidor* de aplicaciones y el resto del sistema, y es quien propiamente "traduce" el *protocolo X* a peticiones de sistema.

Cliente de X. Un *cliente* envía requisiciones de dibujo e información al *servidor*, y éste responde a la entrada, responde las requisiciones de información y envía reportes de error.

El manejador de ventanas. Por ejemplo, es un *cliente* que tiene autoridad sobre la apariencia de las ventanas en el monitor o unidad de despliegue.

El sistema de ventanas X. No está limitado a un sólo *cliente* interactuando con un solo *servidor*. Puede haber varios *clientes* interactuando con un solo *servidor*, que es el caso cuando varias aplicaciones se despliegan en un sólo monitor.

También, un *cliente* puede comunicarse con varios *servidores*, que ocurriría cuando un programa de mensajes despliega lo mismo en la pantalla de varias personas. (Ver figura 10.2).

Después de presentar estos antecedentes en general, en cuanto al *protocolo X*, es importante mencionar, que el *protocolo X*, se usa

relacionándolo con las *terminales X*, utilizando el *sistema operativo UNIX (Solaris)*, o se pueden utilizar diferentes plataformas como:
PC (Personal Computer), MAC (Macintosh), IBM (International Business Machine), NCD (Network Computing Devices), HP (Hewlett Packard) entre otros, utilizando una *terminal X*, el *protocolo X*, y el *protocolo TCP/IP* donde se puedan:

* Correr múltiples aplicaciones en una pantalla.
* Ser parte de una *red*.
* Seguridad
* Comunicarse con varios *servidores*
* Habilidad de aprovechar ventajas que ofrece la interface gráfica de usuarios.

ARQUITECTURA

CLIENTE – SERVIDOR

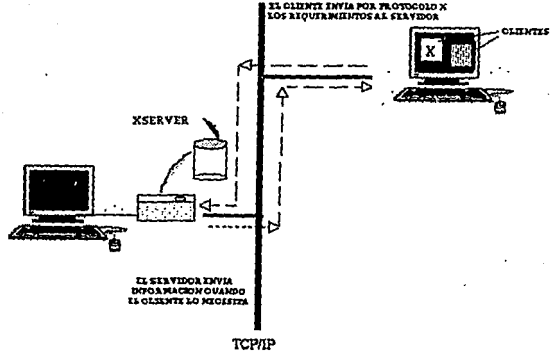


Figura 10.2

Presenta los componentes y funcionamiento de la arquitectura *Cliente-Servidor (xserver y el cliente)*.

10.3 Actualizaciones

Estas dependen particularmente de:

Servidor de X: El fabricante esta constantemente liberando versiones optimizadas de su *servidor de X.*

Librerías y binarios de X11: El consorcio X los va liberando como nuevas versiones. Entre versión y versión pueden existir compendios de mejoras a las librerías.

Sistema operativo: Depende del proveedor de *estaciones de trabajo.*

Procesamiento local. El procesamiento se dedica exclusivamente a las tareas de despliegue, liberando al *servidor* de aplicaciones de esta tarea.

Procesamiento remoto. El trabajo que queda para el procesador del *servidor* de aplicaciones es el de procesar la información de la aplicación y calcular su geometría o posicionamiento, sin ocuparse del despliegue, por lo que la respuesta se vuelve muy rápida.

10.4 Necesidades y beneficios en el uso de terminales X

La funcionalidad de las *terminales X* está orientada a aquellos administradores que necesitan expander el número de usuarios de su *red*, sin que esto implique añadir poder de cómputo o capacidad a un *servidor*, y mucho menos tener un *servidor* extra.

El mayor beneficio que se obtiene de añadir *terminales X* a una *red* es la capacidad en número de usuarios que atenderá el mismo *servidor.* Salvo en tareas muy específicas como diseño mecánico o simulaciones animadas que requieren gran interacción con el *servidor* y mucho procesamiento, la mayoría de los usuarios no satura la capacidad del *servidor.* Puede ocurrir que en un momento dado, varias transacciones

formen un pico en el uso y acceso de la red y de la información, pero generalmente en usuarios estándar esto rara vez se presenta. De modo que el beneficio que nos ofrecen las terminales X es el de sacar mejor provecho de la capacidad de un servidor, que no necesita procesadores extra para liberar carga de trabajo y sin embargo puede aún recibir varios usuarios más.

Para determinar si existe la necesidad del uso de *terminales X*, se puede observar lo siguiente:

Las *terminales ASCII* no tienen la capacidad de manejar un ambiente gráfico, y existen usuarios que requieren de más de una conexión o aplicación simultáneas. El cambio de *terminales ASCII* a *terminales X*, indudablemente aumentará la capacidad y productividad de los usuarios. Con esto se añade capacidad para sesiones múltiples y el ambiente de ventanas, además del acceso a otras aplicaciones vía X y **xterm** o **telnet** simultáneamente a uno o varios *hosts*.

Si con los usuarios existentes se puede determinar que el procesador o procesadores de los *servidores* existentes está subutilizado, no es necesario añadir poder de cómputo sino capacidad de usuarios.

Upgrades. Actualizaciones de *hardware*: La misma terminal puede ir avanzando con el cambio de sistemas. El cambio de *hardware* se hará cuando el usuario requiera una terminal mas rápida o de mayor resolución. Las *terminales X* no se vuelven obsoletas.

Actualizaciones de *software: El servidor de X* se actualiza periódicamente para optimizar nuestras respuestas . Esta actualización se reduce al añadir archivos nuevos, y en un solo cambio se actualizan todas las terminales. Las tareas administrativas se simplifican sensiblemente.

Actualizaciones del sistema: Si se necesita actualizar la versión de *sistema operativo*, librerías o aplicaciones, estos cambios no afectan a las *terminales X*. Estas darán al usuario sus aplicaciones nuevas sin necesidad de hacer cambios.

10.5 Criterios de Evaluación

Precio. Es importante evaluar ventajas/precio. Este varían entre 1,500 y 7,500 USD. Se puede conseguir una *terminal X* modelo *NCD15b* de buena calidad con resolución de 1,024x768 *pixeles*, 16 mhz, monocromática, 2*mb* de memoria, procesador para despliegue *Motorola* 680x0 desde USD 1,500. Existen diferentes modelos que tienen una orientación específica. Las necesidades de cada usuario son diferentes y se ajustan a modelos diferentes para satisfacerlas. algunos modelos *NCD15r, NCD19r, MCX-L.* 10.1

Desempeño. Evalúe con una aplicación real, con el *servidor* que tenga en uso. En este punto hay que considerar las características del *software* del fabricante para instalación y conexión de las terminales.

Línea de productos. Hay que considerar la variedad de opciones que cada fabricante ofrece, de acuerdo al tipo de despliegue y aplicaciones necesarias. En esto se considera si se necesita un monitor monocromático o de color, el tamaño y resolución requerido, y por supuesto, la capacidad de expansión.

Ergonomía. Entre las características que se buscan está la comodidad que va a tener finalmente el usuario. Se puede considerar el trabajo silencioso, las opciones de teclado y la movilidad del monitor y el diseño general, de la *terminal X.*

Opciones de conectividad. Se refiere a que la terminal ofrezca la opción de conexión que se adapte al tipo de *red* en uso, como *token /ring, ethernet,* etc.

NCD, NCDware 3.0 Advanced User's Guide for UNIX® Systems, p. 1.2

10.6 Análisis del *software*

Este es el punto clave en la evaluación, pues en muchos sentidos, es el *software* el que marca las diferencias entre fabricantes. Este muestra la facilidad de uso y acceso de diferentes maneras al *servidor de X*, la forma en que corre, el consumo de recursos y su administración.
Hay que evaluar la *conectividad* que tenga a diferentes plataformas y que soporte varios tipos de *red*, como *TCP/IP*, *DECnet*, *LAT*; los tipos de emulación que se puedan tener y que soporte *accesos remotos*.

Por otro lado, es el *software* el que determina la facilidad de administración de una *red* con *terminales X*, si éstas se pueden configurar de manera remota y el soporte al *protocolo XDMCP* (*X Display Manager Control Protocol*).

11. Archivos de arranque para X11

11. Arranque de X11

11.1 Estructura de archivos

Siguiendo la estructura convencional de *UNIX*, el lugar más adecuado para la localización de los archivos del *protocolo X* son los siguientes:

/tftpboot
/usr/lib/X11
/usr/bin/X11

/tftpboot: En el caso del *servidor* de arranque de X, normalmente ésta es la localización de los archivos de *boot* (arranque). En la mayoría de los casos, la terminal conoce su *dirección Ethernet* (*dirección física*), que de fábrica viene grabada en un *chip PROM*.

De existir un archivo **/etc/ethers** en alguna máquina en la *red*, le puede dar el nombre al que está asociada la terminal y con éste obtener su dirección en el archivo **/etc/hosts**. Con la *dirección Internet* (*dirección lógica*) traducida a hexadecimal, cada terminal puede buscar su archivo de arranque en el directorio ***/tftpboot***.

Si el servidor en uso no tiene el archivo **/etc/ethers**, la primera vez que se arranque la terminal buscará el archivo estándar (**Xncd15b**, que se encuentra bajo el directorio ***tftpboot***) correspondiente a su modelo, y se podrá grabar en la memoria de la terminal la dirección *Internet* correspondiente, para que en ocasiones subsecuentes, conozca su dirección y busque directamente el archivo que le corresponde en ***/tftpboot***.

/usr/lib/X11 : Este es el directorio convencional para las librerías de *X11*. Aunque algunos servidores *UNIX* incluyen la llamada **Xlib** dentro de su estructura **/usr/lib**, éste puede ser un punto para su localización. (Ver figura 11.1) :

Estructura de archivos de arranque para el protocolo X

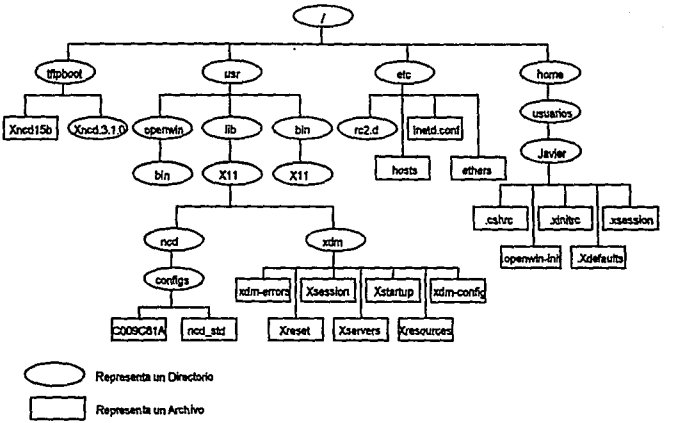


Figura 11.1

Archivos de arranque de X11.

En el caso de que el servidor ofrezca un ambiente gráfico como *Open Look* o *Motif*, bajo el directorio **/lib** de éstos se podrá localizar una o varias librerías llamadas */libX**.

Es necesario tener una variable que indique la ruta de búsqueda (el *path*) para las librerías. Esta variable es **LD_LIBRARY_PATH**,que tiene la ruta **/usr/lib/X11**.

/usr/bin/X11 :Este es el directorio convencional para , los binarios o ejecutables de X (y sus fuentes en algunos casos) pueden ya estar incluídos dentro del servidor de *UNIX*. Normalmente, los archivos contenidos en **/usr/bin/X11** incluyen algunos binarios de X, más no todos, únicamente los más utilizados. Estas fuentes son de dominio público, y se pueden pedir directamente al *MIT* (Massachusetts Institute of Technology, cubriendo los gastos de envío y servicio, o bien, por *red Internet* mediante **ftp**). En algunos sistemas de ventanas, los ejecutables de X (o parte de ellos) están incluidos en el directorio **/bin** bajo la rama del directorio que contenga el ambiente gráfico.

Esto ocurre generalmente en aquellos manejadores de ventanas que soporten *X11*.

Lo más conveniente, es especificar la localidad de los binarios de *X11* en el *search Path* (la ruta de búsqueda), normalmente en el archivo **.cshrc** (C shell) o **.profile** (Bourne shell) o bien, en el archivo **Xsession**. El *path* normalmente ya contendrá la vía de acceso al resto de los binarios del sistema, como **/bin**, **/usr/bin**, **/usr/etc**, etc. La lista de éstos directorios varía de un sistema a otro.

Hay que tomar precauciones con el orden que se le dé, ya que las búsquedas son secuenciales de izquierda a derecha. Por ejemplo, en algunos casos no es conveniente tener "." al principio del *path*, pues usuarios experimentados pueden tomar ventaja de esto.

11.2 Archivos esenciales de X11

Para ofrecer servicios de X, el servidor necesitará de varios archivos que contengan la información de los despliegues que se manejarán, la configuración del ambiente gráfico, y algunos datos más, necesarios para la administración de *X11*.

Un proceso que facilita esto es el **xdm** (X Display Manager), y la localización es bajo (Ver figura 1)

/usr/lib/X11/xdm

Estos archivos también le dan al administrador la oportunidad de definir opciones particulares a su sistema para todas sus terminales en su *red*. En algunos casos, la existencia de estos archivos es sólo para mantener la compatibilidad con **xdm** bajo *X11R3* (revisión tercera de *X11*) y otros pueden no existir y el sistema tomará los valores por default.

11.2.1 El Directorio xdm y sus archivos

Xreset. Este archivo se utilizaba hasta la versión *X11R3,* contenía comandos para reinicializar X al terminar una sesión pero antes de cerrar el despliegue. Estos comandos generalmente se *redu*cían a una llamada al archivo **Xstartup**.

Xresources. Todas las características de *X11* son configurables. En este archivo se determinan colores, tipos de letras, mensajes, señales, características del cursor, y muchas opciones más.

Xservers. Hasta *X11R3,* el proceso **xdm** necesitaba que se especificara en este archivo todos los servidores de despliegue que tuvieran posibilidad de conectarse. En *X11R4*, y ahora *X11R5*, este archivo no es necesario.

Xsession. Este archivo va a determinar el ambiente de trabajo. En *UNIX* estándar, al entrar un usuario el sistema leerá su archivo **.login** o **.profile.**

En el ambiente X, **Xsession** sustituye a estos archivos. En él se deben contener todas las variables de ambiente que necesita un usuario, y es donde se determina qué otros archivos se leerán. Una opción a este archivo es que cada usuario contenga en su directorio un archivo equivalente llamado **.xsession**. De no existir, se leerá este archivo, que es genérico.

Es en éste archivo donde se indica el manejador de ventanas que se utilizará. Si este archivo no existe, **xdm** no podrá dar inicio a una sesión. La localización para estos archivos es **/usr/lib/X11/xdm**

Xstartup. Este archivo tampoco está en uso a partir de *X11R4*. Su contenido eran comandos para iniciar la sesión después de validar el usuario.

xdm-config. Este archivo se lee al iniciar **xdm**. En él se especifica la ruta de acceso y los nombres de otros archivos que se leerán. Si se desea utilizar otro nombre, no estándar, en lugar de los mencionados en esta lista, se debe especificar en este archivo.

xdm-errors. El proceso **xdm** que debe estar corriendo en el servidor, posiblemente encuentre algún problema en su operación; éstos y cualquier otro mensaje se almacena en este archivo para poder ser analizado por el administrador.

.xinitrc. La localización natural de este archivo es en el directorio raíz de cada usuario ($HOME). Aunque no es un archivo de distribución de **X** ni **xdm**, es mencionado porque contiene las preferencias de cada usuario para iniciar su sesión de X cuando no se está utilizando **xdm**. (Ver figura 1)

11.3 Manejadores de Ventanas

El manejador de ventanas que se vaya a utilizar, es transparente para la utilización de **X11**. Existen muchos manejadores de ventanas, todos ellos basados en X, y entre ellos, los más populares son *Motif, Open Look, Open Desktop.* Además, en algunas terminales X está disponible un manejador de ventanas local, tipo *Motif.*

El desempeño de las aplicaciones y de la misma terminal es independiente del manejador de ventanas que se vaya a utilizar.

El único punto donde hay que tener cuidado respecto al manejador de ventanas, es en el archivo **Xsession**, pues es aquí donde se arranca el manejador de ventanas y donde se definen las variables para sus búsquedas.

11.3.1 Tipo de terminal

xterm, es un programa que emula una terminal para poder ejecutar comandos, funciona con los tipos más comunes de el directorio terminfo, incluyendo *'xterm'* , *'vt102', 'vt100' y 'ansi'*. El programa **xterm** busca estos tipos automáticamente en los archivos del directorio terminfo y obtiene el valor para las variables de ambiente *TERM y TERMCAP.*

El valor estándar, *'xterm'*, se distribuye con X y provee operación muy confiable. El archivo se encuentra en los directorios de las cintas de distribución de X y es conveniente copiar esta línea de definición (el archivo se llama termcap también), y añadirla como el primer tipo de terminal que se encuentre en **/usr/share/lib/terminfo** en el sistema.

11.4 Archivos de Arranque para X11

Antes de definir archivos de arranque, es necesario entrar un poco en lo que es **xdm**, que en si es un protocolo que trabaja con *X11*. Como ya se ha dicho, una forma de iniciar la sesión de *X11* es con **xdm**, además de poder iniciarse manualmente: **/usr/bin/X11/xdm** *-c*.

Con la opción *-c*, para leer la nueva configuración sin perder la que ya se tiene.

A continuación se dará una descripción de ***xdm***, y de como se relaciona con sus archivos.

11.4.1 XDM (X Display Manager)

El protocolo del manejador de despliegues de X (**xdm**) es un proceso residente en el *host* que escucha las requisiciones de las *estaciones de despliegue*. Cuando recibe una requisición, **xdm** toma el manejo de la estación de despliegue, donde una ventana de *'login'* o entrada, verificando el *password*, y arrancando una sesión de X.

Los usuarios no necesitan saber los detalles de X, tales como disponer variables para dirigir la salida de sus procesos a la estación apropiada.

En su uso más básico, **xdm** emula un *'login'* en una *estación de despliegue* mientras mantiene el servidor de X corriendo, pide un nombre de usuario y *password*, y maneja una sesión estándar. Esta sesión puede ser adoptada a través de un *script* para todo el sistema llamado **Xsession**, que puede ser usado para disponer variables automáticas e iniciar algunos *clientes* básicos.

En una implementación mas avanzada, los usuarios pueden diseñar sus propias sesiones usando un **~/.xsession** (~, denota el home) individual. El administrador del sistema puede crear archivos especiales para manejar mas de un despliegue a la vez, y disponer recursos de X específicos para sistema o usuarios.

A través de una entrada en un archivo de inicio, el *'demonio'* de **xdm** en el *host* (normalmente durante el arranque del servidor), lee la configuración de **xdm** listando los servidores de X. Una vez que un *servidor de despliegue* es inicializado, **xdm** abre una conexión al *servidor de despliegue* del usuario con una ventana de *login*. El usuario se da de alta y la sesión de X inicia inmediatamente. **xdm** inicia cualquier *cliente* listado en el archivo **.xsession** del usuario.

La versión de **xdm** descrita aquí es para *X11R4*.

X11R4 xdm. Este *demonio* es compatible con su predecesor de *X11R3* y provee un mejor método para establecer conexiones entre servidores y estaciones de trabajo. Se basa en una extensión del protocolo **xdm** llamado **XDMCP** (*X Display Manager Control Protocol*).

XDMCP consiste de los siguientes cinco pasos:

1. Requisición de un Manejador de Despliegue
2. Autentificación del Manejador de Despliegue
3. Despliegue de la ventana de *login*
4. Inicio de la Sesión X
5. Fin de la Sesión X

11.4.2 Manejo de Errores

El demonio de *xdm* lleva una bitácora de errores en el archivo **/usr/lib/X11/xdm/xdm-errors.** Los mensajes en este archivo pueden ayudar en la resolución de fallas. Debe ser monitoreado constantemente porque puede crecer muy rápidamente cuando **xdm** encuentra demasiadas fallas.

11.5 Configuración de clientes, archivos de arranque

A nivel usuario, los archivos de arranque son **Xsession (\$ HOME/.xsession)** bajo **XDM** o **.xinitrc** . Lo más cómodo es dejar que la computadora haga el trabajo, y sin duda, los usuarios querrán correr varios *clientes* de X automáticamente cuando se den de alta. Estos dos archivos son *scripts* de inicialización, y se utilizan en dos situaciones diferentes :

* Si **xdm** está corriendo X, el *script* que busca es **Xsession**.
* Si se inicia la sesión de X con **xinit**, será **.xinitrc** el archivo a leer, y el comando **xinit** debe aparecer al final del archivo **.login**.

xinit normalmente arranca el *servidor* y corre un *cliente* **xterm** (emula una terminal virtual), pero si existe el archivo **.xinitrc** para el usuario, **xinit** arranca el *servidor* y ejecuta **.xinitrc** .

(Ver apéndice A)

En general, se puede decir que, con algunas variaciones dependiendo del ambiente específico, un *script* de arranque debe :

* Disponer las variables de ambiente
* Cargar los recursos adecuados con **xrdb** *(X resources data base)*
* Arrancar el manejador de ventanas
* Iniciar otros clientes que el usuario prefiera por default, como **xterm, xclock**, etc.

El *script* puede ser un *script C shell* o *Bourne shell* indistintamente.

12. Servicios requeridos en la instalación de una terminal X

12. Servicios requeridos en la instalación de una terminal X

La instalación que se describirá en esta tesis, está enfocada a un sistema *UNIX* estándar y a los servicios de *red* que habitualmente se encuentran disponibles en *Sistemas BSD (Berkeley Software Distributions) o SystemV (System Five).*

NOTA: La instalación se llevará a cabo sobre equipo Sun Microsystems utilizando el *software* de X de **Network Computing Devices (NCD).**

12.1 Dirección de la unidad

Dependiendo del fabricante, las *terminales X* pueden soportar varios *protocolos de arranque.* Estos servicios son generalmente llamados desde el archivo **/etc/inetd.conf**, aunque se pueden presentar variaciones. (Ver apéndice A)

El primer paso es determinar la *dirección Internet* que le corresponderá a la unidad. Usando las convenciones locales de la *red*, se debe asignar una dirección acorde al *protocolo Internet*, sobre todo hay que tomar precauciones si se tiene conexión con *SRI-NIC (Stanford Research Institute Network Information Center)*, la organización que administra *Internet*, Internacionalmente.

Cuando se hubo determinado la dirección, hay que añadirla a la base de datos que describe las *direcciones Internet* o *direción lógica* en al archivo **/etc/hosts** .

Por ejemplo, si se va a asignar la dirección 192.9.200.101 a la unidad con nombre "maye20", se debe añadir la línea :

192.009.200.101 maye20

Verificar que existe algún *protocolo* para resolución de direcciones (***BOOTP* (*Bootstrap* Protocol), RARP (Reverse Address Resolution Protocol), MOP (Maintenance Operation Protocol)**) en la *red* local, si no existe se tendrá que iniciar a la *terminal X* manualmente, de la siguiente forma:

bt Xncd19c 192.9.200.101 192.9.200.200

Donde:

bt, es un instrucción propia de la *terminal X*, que le indica su *dirección lógica* y la dirección de su *servidor*, para que no busque el *protocolo* que traduce su *dirección Ethernet* y/o física a la lógica.

192.9.200.101 *Dirección Lógica de la Terminal X.*
192.9.200.200 *Dirección Lógica del Servidor de X.*

Después de la instrucción anterior, guardar los valores en memoria. Normalmente, las *terminales X* tienen una característica llamada **NVRAM** *(non-volatile read access memory)* que puede almacenar éste y otros valores.

Si se está utilizando el servicio de *Yellow pages*, en algunos sistemas conocido como *NIS (Network Information Service)* , será necesario actualizar la base de datos **yp** *(yellow pages)* con los siguientes comandos :

 # cd */var/yp*
 # make *hosts*

12.2 Servicios de *red* requeridos en el *servidor*

Para soportar una unidad en la *red* local, normalmente se requieren tres clases de servicios :

* Determinación de direcciones
* Servicios de *boot* a través de la *red*
* Transferencia de archivos

Para *TCP/IP*, si la *red* está conectada por un *gateway* a otras *redes*, también será necesaria la determinación de máscaras de *subred*.

Es necesario determinar desde un principio el *protocolo* con el que se va a trabajar, teniendo la seguridad que tanto la *terminal X* lo soporte como que el *servidor* pueda ofrecerlo.

12.2.1 Inicio de Servicios desde el archivo inetd.conf

Las utilerías de *red* se inician ya sea en el archivo **/etc/rc#.d** o en **/etc/inetd.conf**. (Ver Figura 12.1)

Si **/etc/inetd.conf** no existe, se puede checar también un archivo llamado **/etc/servers**.

Para sistemas que utilicen *SystemV*, el equivalente al **/etc/rc.local** se encuentra en **/etc**, en los directorios **/etc/rc#.d** (# = 0, 1, 2, 3, 5, S). Bajo estos directorios se localizan varios archivos, y entre todos ellos configuran los servicios que estamos buscando, en particular el directorio **rc2.d** .

Estructura de archivos de arranque para el protocolo X

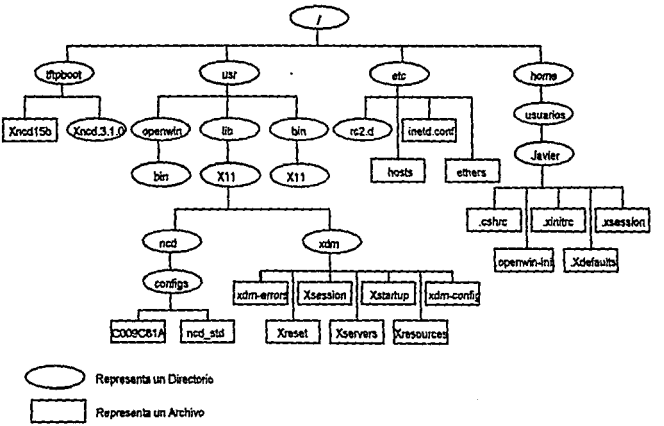


Figura 12.1

Archivos de arranque de X11.

12.2.2 *Protocolos* utilizados para instalar una terminal X

Los *protocolos* (Conjunto de reglas para indicar el modo de comunicación a través de la *red*) que deben estar presentes para instalar una terminal son los siguientes:

XDMCP X Display Manager Control Protocol.

XDM X Display Manager.

RARP Reverse Address Resolution Protocol

BOOTP Bootstrap Protocol.

TFTP Trivial File Transfer Protocol

DNSBM Domain Name Service.

12.2.3 *Protocolo BOOTP* (*Bootstrap Protocol)*

Se puede utilizar este *protocolo* para resolución de direcciones y para especificar los archivos a cargar para *boot.* Para verificar que este "*demonio*" esté configurado, habrá que buscar una línea en el archivo **/etc/inetd.conf** con el siguiente comando :

 # grep bootp /etc/inetd.conf

Dado que la configuración de *bootp* varía de un *host* a otro, es conveniente revisar la documentación que al respecto se tenga. También es necesario revisar la documentación y las instrucciones de la *terminal X* y sus implementaciones de *software* para el acceso a este *protocolo.*

12.2.4 *Protocolo RARP* (Reverse Address Resolution Protocol)

RARP sólo puede ser usado para determinación de direcciones para proveer a la unidad con su *dirección IP*. El *demonio (proceso que inicia rarp)* de *RARP* es generalmente **rarpd** y normalmente ya está configurado. Se puede verificar su configuración en un archivo de arranque del directorio **/etc/rc2.d** de la manera siguiente :

 # grep rarp /etc/rc*

El resultado es el siguiente, nos indica el archivo en este caso **K60nfs.server**, y la línea correpondiente, que levanta *demonio*.

/etc/rc2.d/K60nfs.server: #rarpd ie0 `hostname` ;

Si la línea está comentada como en el ejemplo (para el *shell*, dentro de un archivo, el # sirve para comentarios), es necesario editar **/etc/rc2.d/K60nfs.server** para quitar el # de la línea de comando y después iniciar el *demonio* a mano :

 # in.rarpd -a

El funcionamiento de *RARP* esta orientado a darle a cada unidad su dirección cuando necesita un archivo de *boot*, como en este caso . Para iniciarlo generalmente verifica la existencia de un directorio ***/tftpboot*** al arrancar; si no existe, es conveniente crearlo.

Cuando este servicio está disponible, se debe agregar también una línea en el archivo **/etc/ethers** (normalmente). *La dirección Ethernet* se asigna a cada unidad desde fábrica, y está grabada en un chip en la máquina, de modo que una unidad o estación tiene definida su *dirección Ethernet* 00:00:a7:08:f2:4b que en el archivo **/etc/ethers** se agregará de la forma siguiente:

00:00:a7:08:f2:4b maye20

Y nuevamente, si se utilizan servicios de *yellow pages*, habrá que actualizar las bases de datos correspondientes.

12.2.5 *Protocolo TFTP* (Trivial File Transfer Protocol)

Este *protocolo* se utiliza para cargar el archivo de *boot* de la terminal o para proveer soporte para lectura/escritura en archivos de fonts, el archivo de definición de colores (***/usr/openwin/bin/***rgb.txt) o archivos de configuración remota si están disponibles por el fabricante.

El *demonio* para *TFTP*, **tftpd**, se configura normalmente en **/etc/inetd.conf**.

Por ejemplo :

Se indica el *protocolo* que se utiliza en este caso **tftp** , la ruta donde se encuentra el *demonio* o proceso y se ejecuta.

 # tftp dgram udp nowait /usr/sbin/in.tftpd in.tftpd

Para servidores Sun con *SunOS4.X*, además existe la opción de seguridad que se habilita con el argumento *-s*, llamada "**secure tftp**" Cuando está especificada, la línea generalmente se ve de la siguiente manera :

 # tftp dgram udp wait root /usr/etc/in.tftpd in.tftpd -s *tftpboot*

Es conveniente descomentar esta linea y quitar la opción:

"**-s *tftpboot***"

La opción de seguridad hace que **tftpd** haga un **chroot** al directorio ***tftpboot***, que restringe a *TFTP* a los archivos bajo este directorio.

Lo más conveniente es no utilizar esta opción de seguridad, porque de otra manera todo el *software* provisto para la terminal debe radicar bajo ***tftpboot***, que es normalmente parte de la partición de *root*, y podría

ocasionar que el sistema se llenara, y las ligas a cualquier otro lado no funcionarían.

12.2.6 Servicio de nombres

Si la *red* soporta este servicio, se pueden dar nombres en lugar de direcciones cuando se especifiquen *servidores* en el archivo de configuración para la terminal. También hay que asegurarse de que la terminal soporte este servicio.

Para verificar en el *servidor* si se está ofreciendo este servicio, se puede hacer con :

 # grep named /etc/inetd.conf
name dgram udp wait root *lusr/etc/in.tnamed in.tnamed*

En el caso de que el *demonio* sea ***in.tnamed***, y el servicio *IEN116* (Institute of Electrical an Electronics Engineering)

Para el servicio *DNS* (*Domain Name Service*) se encontraría en ***/etc/rc#.d***

 # grep *named /etc/rc**
if [-f /usr/etc/in.named -a -f /etc/named.boot]; then
/usr/etc/in.named; echo -n ' named') > /dev/console

En servidores Sun, además del ***/etc/named.boot*** se encuentran los archivos **/etc/named.ca** y **/etc/named.db**.

12.3 Instalación del software, para terminales X

La instalación del *software* para las *terminales X* es sencilla, de hecho, la mayoría de los vendedores proveen con el *software* algún *script* de instalación que da opciones al administrador. Entre estas opciones se encuentran el *servidor* de *boot*, el *servidor* de fonts, y la ruta de acceso a los directorios para librerías y binarios. Generalmente, estos *scripts* de

instalación configuran los aspectos básicos para el acceso de las terminales, pero aún queda mucho trabajo por hacer. La función de los *scripts* de instalación de *terminales X*, es únicamente la de instalar el *software* necesario para el arranque de las terminales, de modo que posiblemente no dejen una estructura para los directorios de *X11*.

Es conveniente crear la estructura de *X11* antes de instalar el *software*, y estar seguro de que los servicios de *red* y el *protocolo* para accesarlo estén disponibles y bien configurados, para que al instalar el software no se encuentre con problemas o no indique que la máquina no está preparada para proveer estos servicios.

Como ejemplo, y para efectos de la instalación, se tomará una *terminal X* de *NCD*, modelo *15b*, en un *servidor* Sun.

Será necesario revisar que los servicios de *red* estén disponibles. En este caso, el *protocolo* a utilizar será *TFTP (Trivial File Transfer Protocol)*, de modo que siguiendo las instrucciones de la puntos antes expuestos, habrá que buscar estos servicios en los archivos indicados.

Para cargar el *software* de instalación de la *terminal X*, es necesario ser el *súper usuario*, *root*, para poder crear las estructuras correspondientes.

A continuación se muestra el proceso de instalación de una *terminal X*, siguiendo el procedimiento provisto por *NCD*.

NCD provee su *software* en cintas o *CDROM*. En cualquiera de estos medios, se encuentra un directorio con 24 archivos en formato *tar*, numerados del 0 al 23. Estos archivos contienen por módulos todas las opciones disponibles en el *script* de instalación .

La instalación que se mostrara es para un *servidor* Sun con *Solaris 2.X* (*SunOS 5.1.3*), con servicios de *tftp*, asumiendo que se cuenta con suficiente espacio en disco bajo **/usr**

El único archivo a extraer es 0 (cero), y el *script* de instalación se encarga de extraer el resto.

12.3.1 Procedimiento de instalación

1. Determinar en que directorios se bajará provisionalmente todo el *software*:

frida# **df** *-a*

Filesystem	***kbytes***	***used***	***avail***	***capacity***	***Mounted on***
/dev/sd0a	*29184*	*15243*	*11023*	*58%*	*/*
/dev/sd0g	*434784*	*224479*	*166827*	*57%*	*/usr*
/dev/sd0h	*551588*	*44795*	*451635*	*9%*	*/apps*
/dev/sd0f	*1286667*	*129781*	*1028220*	*11%*	*/home*
/dev/sd0d	*46903*	*145*	*42068*	*0%*	*/var*

frida#

El total de los archivos del 0 al 23 son aproximadamente 80 *MB* (*megabytes*). De estos, lo básico son aproximadamente 12 *MB* bajo **/usr** (para fonts y librerías) y de 1.5 a 3 *MB* en */* (*root*) por cada modelo diferente de terminal a instalar.

2. En este ejemplo, se puede utilizar la *partición* **/home** que tiene 1 *GB*, y crear un directorio temporal donde se bajen estos archivos.

frida# **mkdir** */home/ncd-tmp*
frida# **cd** */cdrom/ncdware_3_1_0/generic*
frida# **ls**

0	*11*	*14*	*17*	*2*	*22*	*4*	*7*
1	*12*	*15*	*18*	*20*	*23*	*5*	*8*
10	*13*	*16*	*19*	*21*	*3*	*6*	*9*

frida# **cp** *0 /home/ncd-tmp*
frida# **cd** */home/ncd-tmp*
frida#

3. Extraer el archivo 0, se obtiene los *scripts* necesarios para la instalación:

frida# **tar** *xvf 0*

x ./COPYRIGHT, 5442 bytes, 11 tape blocks
x ./README, 2156 bytes, 5 tape blocks
x ./SERVER_NAMES, 987 bytes, 2 tape blocks
x ./mkdirhier, 972 bytes, 2 tape blocks
x ./ncdinstall , 56862 bytes, 112 tape blocks
x ./ncdlinkservers, 56862 bytes, 12 tape blocks
x ./.toc, 709 bytes, 2 tape blocks
x ./.groups, 16447 bytes, 33 tape blocks
x ./.servers, 606 bytes, 2 tape blocks
frida#

4. El *script* de instalación es **ncdinstall**. De aquí en adelante, el mismo *script* da ejemplos y directorios por default para la localización de todos los archivos. 10.2

frida#./ncdinstall

Copyright 1988-1993 Network Computing Devices, Inc.

Portions Copyright 1980-1988 Regents of the University of California
Portions Copyright 1987-1992 Thursby Software Systems, Inc.
This software is provided by Network Computing Devices Inc. only pursuant to a license agreement containing instructions on its use, duplication and disclosure. The software contains confidential and proprietary information constituting valuable trade secrets.

Do you wish to continue [n]? y

When questions are asked they will contain a string in square brackets right before the question mark. This is the default response – that is the response that will be assumed if you simply depress return.

Any time a question is asked you may respond with q, quit, Q, or QUIT to terminate this installation script . If you are being asked for a file or directory name and want to use one of these as a response, simply precede it with ./ (e.g. ./q).

10.2 SilVaTech, Manual Ambiente X, p. 18

The distribution media should be loaded and ready in the appropiate drive. Please enter the name of the device in which you have placed the release media when asked below.

Enter the directory name ? /cdrom/ncdware_3_1_0/generic
Just a moment, please.
The NCD server must be placed on a system that supports a minimum set of network utilities. We'll now see if we can determine which network utilities are available. Please be patient...

You should select all of the options you want installed at this time.Since the information gathered so far is saved, you can always come back and install other options later. The installation options are:

1 - server images
2 - SunOS binaries for NCD utilities
3 - RISC ULTRIX binaries for NCD utilities
4 - IBM AIX (RS/6000) binaries for NCD Utilities
5 - HP-UX 8.05 (HP PA-RISC) binaries for NCD Utilities
6 - SunOS bootpd and template bootptab
7 - color database and key definition database
8 - sample remote config files
9 - font server config file
10 - misc fonts
11 - 75dpi fonts
12 - 100dpi fonts
13 - dw75dpi fonts
14 - dw100dpi fonts
15 - Xol fonts
16 - 3-d fonts for PEX
17 - Type1 fonts
18 - Speedo fonts
19 - manual pages
20 - sources for NCD utilities
21 - example files
22 - siedemo data files
23 - SNMP docs and MIB files
all - install all file groups

Please enter the numbers of all options you want installed at this time separated by spaces.

Items [1 2 3 4 5 6 7 8 9 10 11 12 13 14 19 21]? 1 2 7 8 10 11 12 19 21 22 23

You have chosen the following options:

1 - server images
2 - SunOS binaries
7 - rgb.txt and XKeysymDB
8 - sample config files
10 - misc fonts
11 - 75dpi fonts
12 - 100dpi fonts
19 - man pages
21 - example files
22- siedemo data files
23 - SNMP files

Do you want to change any of these choices [y]? n

The following directories will be used for installation:

option	directory
1 - server images	/tftpboot
2 - SunOS binaries	/usr/bin/X11
7 - rgb.txt and XKeysymDB	/usr/lib/X11/ncd
8 - sample config files	/usr/lib/X11/ncd/configs
10 - misc fonts	/usr/lib/X11/ncd/fonts
11 - 75dpi fonts	/usr/lib/X11/ncd/fonts
12 - 100dpi fonts	/usr/lib/X11/ncd/fonts
19 - man pages	/usr/man
21 - example files	./examples—DOES NOT EXIST
22- siedemo data files	/usr/lib/X11/ncd/demo—DOES NOT EXIST
23 - SNMP files	/usr/lib/X11/ncd/snmp—DOES NOT EXIST

One or more of the directories do not exist. As you are asked to specify each directory you can choose to keep the current specification and the you will be asked whether you want it created.

The directory , "/usr/lib/X11/ncd/demo", doesn't exist.
The data files for demostration programs such as siedemo are usually installed in a sub-directory of /usr/lib/X11/ncd. They need only be accessible by the demonstration program running on a host system.

Enter the name of the directory for demo data files [/usr/lib/X11/ncd/demo]?

The directory you entered, "/usr/lib/X11/ncd/demo", doesn't exist.

Do you want it created [n]? y

The directory , "./examples", doesn't exist.

Several example files are provided for configuring and running NCD X terminals.

Enter the name of the directory for examples [./examples]? /usr/lib/X11/ncd/examples

The directory , "/usr/lib/X11/ncd/snmp", doesn't exist.

The SNMP files include MIBs and documentation.

Enter the name of the directory for SNMP files [/usr/lib/X11/ncd/snmp] ?

Do you want it created [n]? y

The following directories will be used for installation:

option	*directory*

1 - server images	*/tftpboot*
2 - SunOS binaries	*/usr/bin/X11*
7 - rgb.txt and XKeysymDB	*/usr/lib/X11/ncd*
8 - sample config files	*/usr/lib/X11/ncd/configs*
10 - misc fonts	*/usr/lib/X11/ncd/fonts*
11 - 75dpi fonts	*/usr/lib/X11/ncd/fonts*
12 - 100dpi fonts	*/usr/lib/X11/ncd/fonts*
19 - man pages	*/usr/man*
21 - example files	*/usr/lib/X11/ncd/examples*
22- siedemo data files	*/usr/lib/X11/ncd/demo*
23 - SNMP files	*/usr/lib/X11/ncd/snmp*

Do you want to change any directory choice[y]? n

The server images selected will be installed into a subdirectory in "/tftpboot". In order to download them links to the files in the subdirectory should be placed in "/tftpboot". This makes the images bootable assuming you have TFTP or MOP set up correctly.

Do you want the server images linked in "/tftpboot" [y]? y

Choose the set of products for which you want downloadable server images installed. Make selections from the table below and enter each

choice separated by spaces or simply enter the word all.

15b - NCD15b
15r - NCD15r (same as NCD19r)
16 - NCD16
16e - NCD16e
19b - NCD19b
19 - NCD19
19r - NCD19r
14c - NCD14c
17c - NCD17c
17cr - NCD17cr (same as NCD19c)
mcx - any MCX product (same as NCD19c)
19c - NCD19c

19g - NCD19g (same as NCD19c)
all - all servers

List the products for which you want servers installed [15b 15r 16 16e 19b 19 19r 14c 17c 17cr mcx 19c 19g] ? 15b 15r 17cr mcx 19c

You have chosen to install servers for the following terminal products:

NCD15b
NCD19c
NCD19r

Do you want to change any of these choices [y]? n
MOP loadable server images are used with the ULTRIX utility mop_mom.
This utility appears to be either not present or not running on this system.

The TFTP loadable server images are to be installed.
Do you also want to install the MOP loadable server images [n]? n

Do any of tour NCD terminals have PCMCIA Memory Cards installed [n]? n
===== Installing server images ...
===== Done.

===== Installing SunOS binaries ...
===== Done.

===== Skipping ULTRIX binaries ...

===== Skipping IBM AIX (RS/6000) binaries ...

===== Skipping HP-UX (HP PA-RISC) binaries ...

===== Skipping bootp files ...

===== Installing rgb.txt and XKeysymDB ...
===== Done.

===== Installing sample config files ...
===== Done.

===== Installing ncdfs config file ...
===== Done.

===== Installing misc fonts ...
===== Done.

===== Installing 75dpi fonts ...
===== Done.

===== Installing 100dpi fonts ...
===== Done.

===== Skipping dw75dpi fonts ...

===== Skipping dw100dpi fonts ...

===== Skipping Xol fonts ...

===== Skipping PEX fonts ...
===== Skipping Type1 fonts ...

===== Skipping Speedo fonts ...

===== Installing man pages ...
===== Done ...

===== Skipping sources ...

===== Installing examples files ...
===== Done ...
===== Installing siedemo data files ...
===== Done ...
===== Installing SNMP files ...

===== Done ...

For the NCD 19c there are multiple servers installed. The list below display which servers are available. Please select the key for the one you want as the default server for NCD 19c terminals.

Key	*Descripcion*
x	*Standard server with no options*
p	*PEX server with no other options*

Enter the key for the default server for NCD19c terminals? x

frida#

El *script* de instalación copia los archivos de arranque de las terminales bajo **/tftpboot/Xncd.3.1.0**, y crea las ligas necesarias para su localización.

También instala los fonts y binarios seleccionados bajo **/lib/X11** y **/bin/X11.**

El directorio **/lib/X11/xdm** no se instala por default, pero esta contenido en el grupo "*examples*", que fue copiado bajo **/lib/X11/ncd/examples**. Ahí se localiza un subdirectorio llamado **xdm**, que contiene los archivos que ordinariamente se utilizan, y el archivo **Xsession** tiene lo mínimo necesario. Si desea un archivo genérico para *Open Windows* , como es este caso, bajo examples se encuentra también un subdirectorio **openwin**, donde hay un archivo llamado **smpl.xsession**. Este archivo habría que copiarlo al directorio **/lib/X11/xdm** con el nombre de **Xsession** con permisos de ejecución.
Después del mensaje de *Copyright*, se dan algunas instrucciones respecto a la forma de responder a las preguntas del *script*.
Enseguida, el *script* verificará que el usuario sea root. Si no puede determinar esto, lo confirmará con el usuario. También serán verificadas las utilerías de red requeridas para soportar a la unidad *NCD* en la *red*.

Puede ocurrir que las utilerías existan, pero no estén activas. Si se reporta que no se tienen las utilerías necesarias para soportar unidades locales, habrá que determinar si estas utilerías estarán activas cuando las unidades *NCD* estén en uso. Si el *script* supone que no se pueden cargar las unidades desde el sistema donde se está trabajando, preguntará si se desea continuar. Si la respuesta es "no", el *script* terminará.

El *servidor* de cada modelo de *terminal X* es diferente. En algunos casos puede ser el mismo, pero en general, cada modelo de *terminal X* tiene un *servidor* diferente. Con *servidor*, en este caso no se refiere a la máquina que le prestará servicios y procesamiento, sino en el concepto *cliente-servidor* de *X11*, donde el cliente es una aplicación que corre en cualquier unidad de despliegue (incluso localmente) y el servidor es aquél programa que le dará la comunicación a través de *protocolo X* y que controlará los dispositivos de entrada y salida.

De modo que el *servidor de X*, en este caso, es llamado **"Xncd15b"** por ser para el modelo *15b*, y se le llama "*server image*".

La ubicación de estos servidores (server images) se determina buscando la presencia de *TFTP*, para transferir los archivos de fonts.

En este caso, se utilizará *TFTP* la localidad para estos archivos debe ser **/tftpboot.** Para esta instalación, hay que estar completamente seguros de que hay espacio suficiente en la partición de root. No es posible crear ligas a otros file systems para añadir espacio ya que se trata de la partición de root y el directorio para que otras máquinas clientes encuentren sus archivos de boot.

Por cada server image, se deben considerar de 1 a 2 *MB* de espacio disponible en la partición de root.

ncdinstall copiará estos archivos en el directorio */tftpboot*, creando un directorio para cada modelo. La cinta para cada terminal contendrá varias opciones de *servidores*. La variación básica es para *mwt* (multi-window terminal), o el *servidor de X*.

El *hardware* para estas dos opciones es el mismo, pero una *mwt* es un paso intermedio entre una *terminal ASCII* y una *terminal X*. Esta opción es

útil para aquellos usuarios de aplicaciones de modo texto, sin aplicaciones gráficas para aprovechar el ambiente gráfico de X pero que bien pueden aprovechar la capacidad de tener múltiples sesiones abiertas al mismo tiempo.

Cuando estos usuarios tienen capacidad gráfica, todo lo que tienen que hacer es bajar el *servidor* para X y su terminal seguirá siendo la misma, pero usando ambiente gráfico.

A continuación, el *script* iniciará la instalación de los fonts. Para esta instalación, es necesario que ya existan los directorios para los fonts.

En este tipo de *estaciones de trabajo*, se va a utilizar *NFS* (Network File System). Este protocolo no es indispensable, pero en el caso del servidor Sun y de muchos otros, el protocolo está disponible y es conveniente utilizarlo para proveer soporte de lectura/escritura para los archivos de fonts, el archivo de definición de color, archivos de configuración remota y para proveer soporte de escritura para el archivo de "**log**" de los diagnósticos (particular para *NCD*).

Dependiendo de cómo se esté utilizando *NFS* en la unidad *NCD*, es necesario proveer puntos de montaje exportados para estos archivos. Las localidades por default para ellos es :

Configuración remota	**/usr/lib/X11/ncd/configs**
Fonts	**/usr/lib/X11/ncd**
Definición de color	**/usr/lib/X11/ncd/rgt.txt**

Como se utilizará *NFS*, estas son las localidades donde la unidad *NCD* buscará esos archivos. Por lo demás, los archivos binarios y el resto de las librerías se pueden instalar donde sea (aunque es preferible seguir las localidades recomendadas) manteniendo estas localidades especificadas en los archivos de configuración, de recursos y las variables de ambiente.

Con esto termina el *script* de instalación de *NCD*. El resto de la cinta contiene ejemplos, utilerías básicas y fuentes para aquellos procesadores no soportados en los binarios básicos.

12.4 El contenido de la cinta o cdrom es el siguiente

Xncd.3.0	*Servidores* (en forma binaria o ejecutable) para un modelo determinado.
binSun3	Utilerías para fonts y otras utilerías en forma binaria para
binSun4	las siguientes plataformas de *hardware*:
binRiscUltrix	Sun-3 (compilado bajo *SunOS 4.1*)
binVaxUltrix	Sun-4 (compilado bajo *SunOS 4.0.3*)
	DEC RISC ULTRIX (compilado bajo *ULTRIX 4.0*)
	DEC VAX ULTRIX (compilado bajo *ULTRIX 4.1*)
examples	Ejemplos de archivos e instrucciones pertenecientes al ambiente X de un usuario, y los archivos fuente para *SNMP MIB* y *bootp*.
src	Código fuente para muchos programas útiles, como *clientes* locales, manejador de ventanas, utilerías para fonts, etc.
man	Páginas de manual para las utilerías de *NCD*.
ncd	El archivo de definición de color, **rgb.txt**, y directorios de fonts.
ncdinstall	*script* de instalación para cintas de *NCD*.

El contenido del resto de los directorios, se puede copiar a las localidades donde el administrador lo prefiera. Muchos de ellos son simplemente ejemplos básicos, y hay que considerar que como se soportan varias plataformas, los binarios incluidos se encuentran varios directorios.

Básicamente, sin importar la *arquitectura* de que se trate, los archivos de *servidor*, *server images*, se pueden instalar en cualquier tipo de sistema siempre y cuando ofrezca los servicios de que se han hablado. Los directorios para binarios que incluyen éstas cintas son sólo una pequeña parte de los *binarios de X*. Todos estos binarios es preferible tenerlos de las fuentes para X, ya que estarán más completos. *NCD* ofrece una parte de ellos que son las *utilerías básicas de X*.

12.5 Configuración de archivos para *terminales X*

Una vez que se ha instalado el *software* , se puede completar la configuración de las *terminales X.* Como se mencionó anteriormente, el *software* está distribuido principalmente en tres áreas:

/tftpboot
/usr/lib/X11
/usr/bin/X11

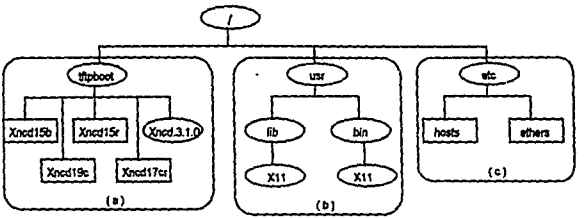
En el directorio **/tftpboot** , se pueden ver los siguientes archivos y directorios (Ver figura 12.2):

```
frida# cd /tftpboot
frida# ls
.          Xncd.3.1.0      Xncd15r         Xncd19c
. .        Xncd15b         Xncd17cr

frida#
```

Xncd.3.1.0 es el directorio que contiene las "imágenes" de los **servidores.** Los demás archivos son ligas duras (*hard links*) a los archivos contenidos en este directorio.

Configuración de Archivos para Terminales X



En el recuadro (a, b), se encuentra distribuido el Software de NCD, provisto para las Terminales X.

En el recuadro (a), se encuentra el directorio tftpboot, contiene las imágenes de los servidores. Los archivos contenidos en este directorio, proporcionará el Xserver a las terminales que corresponda a cada modelo.

Representa un Directorio

Representa un Archivo

En el recuadro (b), se encuentra las librerías y binarios propios de X11, que el usuario puede utilizar, para modificar su ambiente de trabajo.

En el recuadro (c), se muestran los archivos a modificar para que el servidor reconozca a las terminales.

Figura 12.2

Distribución del software y archivos de configuración.

Cada uno proporcionará el *Xserver* a las terminales que correspondan a cada modelo.

Cuando se desea especificar mejor el archivo del cual debe arrancar determinada terminal, se puede hacer una liga con su *dirección Internet* convertida a hexadecimal.

Por ejemplo:

Dirección Ethernet (física) : 00:00:a7:08:f2:4b
Dirección IP (lógica) : 192.9.200.101

Conversión :	decimal	hexadecimal
	192	C0
	9	09
	200	C8
	101	65

Con estos valores, se puede crear un archivo que exclusivamente accesará la terminal con la *dirección IP* mencionada.

El nombre del archivo entonces será la unión de los cuatro pares de números en hexadecimal : *C009C865* . Suponiendo que esta terminal es modelo *19c*, quedará:

frida# **ln** *-s Xncd.3.1.0/Xncd19c C009C865*
frida# **file** *C009C865*
C009C865: symbolic link to Xncd.3.1.0/Xncd19c
frida#

Esto nos asegura que la terminal encontrará su archivo al arranque.

Además, hay que añadir una entrada al archivo **/etc/hosts** y al archivo **/etc/ethers** (para una mejor localización Ver figura 2):

En **/etc/ethers** :

00:00:a7:08:f2:4b maye20

En **/etc /hosts** :

192.9.200.101 maye20

El siguiente punto es configurar el directorio de librerías. El *script* de instalación ha creado el directorio **/usr/lib/X11**, y bajo el, se encuentra lo siguiente:

frida# **cd** */usr/lib/X11*

frida# **ls**
ncd

frida# **cd** *ncd*

frida# **ls**

XKeysymDB	***demo***	***fonts***	***snmp***
configs	***examples***	***rgb.txt***	***src***

frida#

Bajo el directorio **/usr/lib/X11/ncd/examples**, se encuentra un subdirectorio llamado **xdm**, que se puede mover completamente al nivel **/lib/X11** :

frida# **cd** *examples*
frida# **ls**

NCD-Articles	***configs***	***ncdlauncher***	***termcap***	***xmodmap***
ProblemReport	***mwm***	***openwindows***	***twn***	***bootp***
named	***rs6000***	***xdm***		

frida# **cp** *-pr xdm /usr/lib/X11*

frida# **ls** *openwindows*

INSTALL-HELP	***fonts.alias.cm***	***openwin-menu***	***smpl.profile***
README	***openwin-init***	***smpl.cshrc***	***smpl.xsession***

frida# **cp** *smpl.xsession /usr/lib/X11/xdm/Xsession*

frida# **cd** */lib/X11/ncd*

frida# **rm** *-r examples*
frida#

(Ver figura 12.3)

Configuración de Archivos para Terminales X

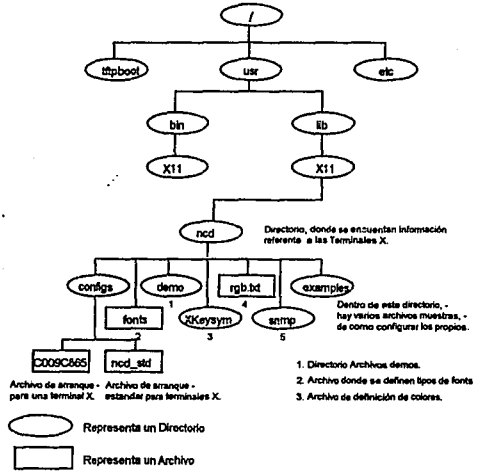


Figura 12.3

Estructura del directorio de librerías para *X11*.

12.5.1 Configuración específica para una *terminal X*

En cuanto a la configuración específica de cada una de las terminales, bajo el directorio **/usr/lib/X11/ncd/configs** se puede crear archivos particulares a cada una, o bien, un archivo genérico que contenga información útil para cada una de las terminales.

Los nombres de éstos archivos se dan igualmente con la *dirección Internet* convertida a hexadecimal, de la forma en que se hizo con el archivo de arranque.

Dirección Internet:

192.9.200.101

Conversión a hexadecimal:

C009C865

Esta configuración se puede escribir desde un menú de las terminales, de modo que es suficiente con crear un archivo vacío con permisos de escritura.

El archivo genérico para configuración de terminales, en este caso específico , es llamado **ncd_std**. Lo único que hay que tener cuidado de no incluir en este archivo, son direcciones y nombres particulares a la terminal, porque de otra manera, se desconfigura la *interface de red*.

El contenido de estos archivos de configuración va de acuerdo a las preferencias particulares de cada usuario, y es completamente dependiente del fabricante. En el caso de las terminales *NCD*, llevan el nombre de la *dirección IP* en hexadecimal (como en **/tftpboot**). La sintaxis de este archivo la específica el fabricante y en algunos casos, hay una opción de "escritura automática".

Apéndice A
Archivos de arranque para X11

Apéndice A. Archivos de Arranque para X11.

El presente Apéndice, tiene por objetivo mostrar los archivos más importantes para , el arranque de *X11*, están escritos bajo el *shell* de *UNIX*, debido a lo cuál sólo se incluyen, como punto de referencia.

Archivo Xsession :

```
#!/bin/sh
#
# Xsession
#
# This is the program that is run as the client for the display
manager.
# This example is quite friendly as it attempts to run a per-user
# .xsession file instead of forcing a particular
# session layout
#
if [ -f $HOME/ .xsession ]; then
    if [ -x $HOME/ .xsession ]; then
        exec $HOME/ .xsession
    else
        exec /bin/sh $HOME/.xsession
    fi
else
    mwm &
    exec xterm -fn 10x20 -geometry 80x24+20+20 -ls
fi
```

Archivo .openwin-init:

```
#! /bin/sh
# @(#)openwin 23.25 90/06/19
if [ -x $HOME/ .openwin-init ]; then # OW user
    OPENWINHOME=${OPENWINHOME:-"/usr/openwin"}
    export OPENWINHOME
    XPATH=/usr/bin/X11:/usr/local/bin/X11
    OWPATH=$OPENWINHOME/bin: \
                $OPENWINHOME/bin/xview
    PATH=$XPATH:$OWPATH:$PATH
    FONTPATH=$OPENWINHOME/lib/fonts: \
            /usr/lib/X11/ncd/fonts
    FRAMEBUFFER=/dev/fb
    HELPPATH=$OPENWINHOME/lib/help
    HOSTTYPE=`arch`
    LD_LIBRARY_PATH=$OPENWINHOME/lib: \
            /usr/lib:/usr/lib/X11
    MANPATH=$OPENWINHOME/share/man:/usr/man
    NEWSSERVER=`$OPENWINHOME/bin/newsserverstr`
    WINDOW_PARENT=/dev/win0
    WMGR_ENV_PLACE_HOLDER=/dev/win3
    XAPPLRESDIR=$OPENWINHOME/lib/X11/app-defaults
    export XPATH OWPATH PATH FONTPATH \
     FRAMEBUFFER HELPPATH HOSTTYPE \
     LD_LIBRARY_PATH MANPATH NEWSSERVER \
     WINDOW_PARENT WMGR_ENV_PLACEHOLDER \
     XAPPLRESDIR
    TMPINIT="$HOME/ .tmp.openwin-init"
     # Openlook Window Manager :
    /usr/openwin/bin/olwm -3 -display $DISPLAY &
    OWINIT=$HOME/ .openwin-init
    # because everything in ow-init is &'d, if we don't  'wait'
    #   here the X session will end
    wait
    rm -f $TMPINIT
    exit 0
```

Continúa . . .

Archivo .openwin-init:

```
else   # generic X
     OPENWINHOME=${OPENWINHOME:-"/usr/openwin"}
     export       OPENWINHOME
     XPATH=/usr/bin/X11:/usr/local/bin/X11
     OWPATH=$OPENWINHOME/bin: \
          $OPENWINHOME/bin/xview
     PATH=:$XPATH:$OWPATH:$PATH
     LD_LIBRARY_PATH=$OPENWINHOME/lib:
          /usr/lib/X11
     export PATH  XPATH OWPATH LD_LIBRARY_PATH
     exec xterm -geometry 80x24+10+10 +ls
fi
```

Archivo .xinitrc:

```
# .xinitrc - OpenWindows startup script.

if [ -f $HOME/ .Xdefaults ]; then
   xrdb $HOME/ .Xdefaults &     #user's resource database
else
   xrdb $OPENWINHOME/lib/Xdefaults &
fi

$OPENWINHOME/lib/openwin-sys &     #system initialization

# Install function key "F1" as an Open Look "Help" key
# This precludes its use by applications
# If your applications use F1 for anything else, comment out the
#following line
xmodmap -e 'keysym F1 = Help'

eval `svenv -env`                  # Sun View binary compatibility

olvwm -3 &                 # OpenLook Window Manager(3-D look)
if [ -x $HOME/ .openwin-init ]; then
   $HOME/ .openwin-init               # Custom OpenWindows tools
else
   $OPENWINHOME/lib/openwin-init      # Default file
fi

wait
```

Archivo: .cshrc

```
# @(#)Cshrc 1.6 91/09/05 SMI
#########################################################
#########
#
#       .cshrc file
#
#       initial setup file for both interactive and noninteractive
#       C-Shells
#
#########################################################
#########

setenv CLARITYHOME /apps/clarity
setenv QUORUM_FLOPPY_PATH none
setenv LD_LIBRARY_PATH
/usr/lib:/usr/openwin/lib:/usr/lib/X11/lib/X11/include
set mychoice=openwin

# add directories for local commands
set lpath = ( )
if ( ${?mychoice} != 0 ) then
      if ( ${mychoice} == "openwin" ) then
            set lpath = ( /usr/openwin/bin/xview /usr/openwin/bin
                        /usr/openwin/lib/libolgx.so.3 $lpath )
      endif
endif

set path = (. ~ $lpath ~/bin /bin/X11 /usr/local/bin /usr/ucb /usr/bin
                  /usr/etc /usr/quorum/bin )

#       cd path
```

Continúa . . .

Archivo: .cshrc

```
#       set this for all shells

set noclobber
setenv OPENWINHOME /usr/openwin
setenv LD_LIBRARY_PATH /usr/lib:/usr/openwin/lib

#       aliases for all shells

alias cd        'cd \!*;echo $cwd'
alias pwd       'echo $cwd'
#umask 002

#       skip remaining setup if not an interactive shell

if ($?USER == 0 || $?prompt == 0) exit

#       settings  for interactive shells

set history=40
set ignoreeof
set prompt="`whoami`# "

#       commands for interactive shells


#       other aliases

#alias a        alias
#alias h        'history \!* | head -39 | more'
#alias u        unalias

#alias
#alias list     cat
#alias lock     lockscreen
#alias m        more
```

Continúa . . .

Archivo: .cshrc

```
#alias type       more
#alias .          'echo $cwd'
#alias ..         'set dot=$cwd;cd ..'
#alias ,          'cd $dot '

alias dir         ls
#alias pdw          'echo $cwd'
#alias la          'ls -a'
#alias ll          'ls -la'
alias ls          'ls -Fa'


alias bye         logout


alias openwin     /usr/openwin/bin/openwin
alias bye   exit
alias c             '/usr/games/canfield; ls -l; file *'
alias lsl     'ls -l'
alias f       /usr/games/fortune
alias curso         'cd $HOME/rapfiles/cursos/acetatos'
alias word          /apps/quorum/bin/MicrosoftWord
alias excel         /apps/quorum/bin/MicrosoftExcel
alias conf          "cd /usr/lib/X11/ncd/configs; echo `pwd`"

/usr/openwin/bin/xmodmap -e "clear Mod1" -e "add Mod1 =
Linefeed"
/usr/openwin/bin/xmodmap -e "clear Mod5" -e "add Mod5 =
Alt_R"
```

Archivo: inetd.conf

```
#
#ident  "@(#)inetd.conf 1.15  93/08/27 SMI"  /* SVr4.0 1.5*/
# Configuration file for inetd(1M).  See inetd.conf(4).
#
# To re-configure the running inetd process, edit this file, then
# send the inetd process a SIGHUP.
#
# Syntax for socket-based Internet services:
#  <service_name> <socket_type> <proto> <flags> <user>
<server_pathname> <args>
#
# Syntax for TLI-based Internet services:
#
#  <service_name> tli <proto> <flags> <user> <server_pathname>
<args>
#
# Ftp and telnet are standard Internet services.
#
ftp     stream  tcp     nowait  root    /usr/sbin/in.ftpd in.ftpd
telnet  stream  tcp     nowait  root    /usr/sbin/in.telnetd
        in.telnetd
#
# Tnamed serves the obsolete IEN-116 name server protocol.
#
name    dgram   udp     wait    root    /usr/sbin/in.tnamed
        in.tnamed
#
# Shell, login, exec, comsat and talk are BSD protocols.
#
shell stream    tcp     nowait  root    /usr/sbin/in.rshd
        in.rshd
login stream    tcp     nowait  root    /usr/sbin/in.rlogind
        in.rlogind
```

Continúa . . .

Archivo: inetd.conf

```
exec stream     tcp  nowait     root  /usr/sbin/in.rexecd
      in.rexecd
comsat   dgram      udp  wait  root  /usr/sbin/in.comsat
      in.comsat
talk  dgram      udp  wait  root  /usr/sbin/in.talkd      in.talkd

# Must run as root (to read /etc/shadow); "-n" turns off logging in
utmp/wtmp.
#
uucp stream     tcp  nowait     root  /usr/sbin/in.uucpd
      in.uucpd
#
# Tftp service is provided primarily for booting.  Most sites run
this
# only on machines acting as "boot servers."
#
#tftp dgram      udp  wait  root  /usr/sbin/in.tftpd      in.tftpd -s
/tftpboot
#
# Finger, systat and netstat give out user information which may
be
# valuable to potential "system crackers."  Many sites choose to
disable
# some or all of these services to improve security.
#
finger      stream     tcp  nowait     nobody
      /usr/sbin/in.fingerd  in.fingerd
#systat     stream     tcp  nowait     root  /usr/bin/ps      ps -ef
#netstat    stream     tcp  nowait     root  /usr/bin/netstat
      netstat -f inet
#
# Time service is used for clock synchronization.
#
```

continúa . . .

Archivo: inetd.conf

```
time stream     tcp  nowait     root  internal
time dgram      udp  wait root  internal
#
# Echo, discard, daytime, and chargen are used primarily for
testing.
#
echo stream     tcp  nowait     root  internal
echo dgram      udp  wait root  internal
discard    stream     tcp  nowait     root  internal
discard    dgram      udp  wait root  internal
daytime    stream     tcp  nowait     root  internal
chargen    stream     tcp  nowait     root  internal

# RPC services syntax:
# <rpc_prog>/<vers> <endpoint-type> rpc/<proto> <flags>
<user> \
#  <pathname> <args>
#
# <endpoint-type> can be either "tli" or "stream" or "dgram".
# For "stream" and "dgram" assume that the endpoint is a socket
descriptor.
# <proto> can be either a nettype or a netid or a "*". The value is
# first treated as a nettype. If it is not a valid nettype then it is
# treated as a netid. The "*" is a short-hand way of saying all the
# transports supported by this system, ie. it equates to the
"visible"
# nettype. The syntax for <proto> is:
#       *|<nettype|netid>|<nettype|netid>{[,<nettype|netid>]}
# For example:
# dummy/1       tli     rpc/circuit_v,udp       wait  root
      /tmp/test_svc    test_svc
```

Continúa . . .

Archivo: inetd.conf

```
#
# System and network administration class agent server
#
# This is referenced by number because the admind agent is
needed for the
# initial installation of the system. However, on some preinstalled
systems
# the SNAG packages may not be present. Referencing the
service by number
# prevents error messages in this case.
#
100087/10 tli     rpc/udp     wait root /usr/sbin/admind admind
#
# Rquotad supports UFS disk quotas for NFS clients
#
rquotad/1  tli     rpc/datagram_v wait root /usr/lib/nfs/rquotad
      rquotad
#
# The rusers service gives out user information.  Sites concerned
# with security may choose to disable it.

rusersd/2-3       tli     rpc/datagram_v wait root
/usr/lib/netsvc/rusers/rpc.rusersd       rpc.rusersd
#
# The spray server is used primarily for testing.
#
sprayd/1   tli     rpc/datagram_v wait root
/usr/lib/netsvc/spray/rpc.sprayd rpc.sprayd
#
# The rwall server allows others to post messages to users on
this machine.
#
walld/1            tli     rpc/datagram_v wait root
/usr/lib/netsvc/rwall/rpc.rwalld   rpc.rwalld
```

Continúa . . .

Archivo: inetd.conf

```
# Rstatd is used by programs such as perfmeter.
#
rstatd/2-4  tli  rpc/datagram_v wait root
/usr/lib/netsvc/rstat/rpc.rstatd rpc.rstatd
#
# The rexd server provides only minimal authentication and is
often not run
#rexd/1        tli  rpc/tcp wait root /usr/sbin/rpc.rexd    rpc.rexd
#
# rpc.cmsd is a data base daemon which manages calendar data
backed
# by files in /var/spool/calendar
100068/2-4dgram      rpc/udp     wait root
/usr/openwin/bin/rpc.cmsd rpc.cmsd
# Sun ToolTalk Database Server
#
100083/1   stream  rpc/tcp wait root
/usr/openwin/bin/rpc.ttdbserverd rpc.ttdbserverd
tftp  dgram      udp  wait  root  /usr/sbin/in.tftpd      in.tftpd -s
/usr/tftpboot
it      root  /export/home/falco/Zync/zyncd
```

Archivo: K95xdminit

```
# xdm startup
state=$1
set `who -r`
case $state in
'start')
        if [ -f /usr/openwin/bin/xdm ]; then
                echo "xdm starting."
                /usr/openwin/bin/xdm -config /usr/lib/X11/xdm/xdm-
config
        fi
        ;;
'stop')
        PID=`/usr/bin/ps -ef | grep /usr/openwin/bin/xdm | awk '{
print $2 }'`
        if [ ! -z "$PID" ]; then
                /usr/bin/kill ${PID} 1>/dev/null 2>&1
        fi
        ;;
*)
        echo "Usage: /etc/init.d/xdminit { start | stop }"
        ;;
esac
exit 0
```

Archivo: .login

```
# @(#)Login 1.14 90/11/01 SMI
##########################################################
##########
#
#       .login file
#
#       Read in after the .cshrc file when you log in.
#       Not read in for subsequent shells.  For setting up
#       terminal and global environment characteristics.
#
##########################################################
##########

#       terminal characteristics for remote terminals:

#       Leave lines for all but your remote terminal commented
#       out (or add a new line if your terminal does not appear).

if ($TERM != "sun") then
#eval `tset -sQ -m dialup:?925 -m switch:?925 -m dumb:?925
$TERM`
#eval `tset -sQ -m dialup:?h19 -m switch:?h19 -m dumb:?h19
$TERM`
#eval `tset -sQ -m dialup:?mac -m switch:?mac -m dumb:?mac
$TERM`
#eval `tset -sQ -m dialup:?vt100 -m switch:?vt100 -m
dumb:?vt100 $TERM`
#eval `tset -sQ -m dialup:?wyse-nk -m switch:?wyse-nk -m
dumb:?wyse-nk $TERM`
#eval `tset -sQ -m dialup:?wyse-vp -m switch:?wyse-vp -m
dumb:?wyse-vp $TERM`
endif
```

Continúa . . .

Archivo: .login

```
#       general terminal characteristics

#stty -crterase
#stty -tabs
#stty crt
stty erase '^h'
#stty werase '^?'
#stty kill '^['
#stty new

#       environment variables

#setenv EXINIT 'set sh=/bin/csh sw=4 ai report=2'
#setenv MORE '-c'
#setenv PRINTER lw
setenv LD_LIBRARY_PATH /usr/openwin/lib
#       commands to perform at login

#w       # see who is logged in

#
# If possible, start the windows system.  Give user a chance to
bail out
#
if ( `tty` != "/dev/console" || $TERM != "sun" ) then
        exit  # leave user at regular C shell prompt
endif

if ( ${?OPENWINHOME} == 0 ) then
        setenv OPENWINHOME /usr/openwin
endif
if (set $OPENWINHOME/bin/openwin ) then
        set mychoice=sunview
endif
```

Continúa . . .

Archivo: .login

```
echo ""
#click -n    # click -n turns off key click
# echo ""
# switch( $mychoice )
# case      openwin:
#      unset mychoice
#      echo -n "Starting OpenWindows (type Control-C to
interrupt)"
#      sleep 5
#      $OPENWINHOME/bin/openwin
#      clear_colormap # get rid of annoying colourmap bug
#      clear        # get rid of annoying cursor rectangle
#      echo -n "Automatically logging out (type Control-C to
interrupt)"
#      sleep 5
#      logout              # logout after leaving windows system
#      breaksw
#      #
#case       sunview:
#      unset mychoice
#      echo -n "Starting SunView (type Control-C to interrupt)"
#      sleep 5
#      # default sunview background looks best with pastels
#      sunview
#      clear        # get rid of annoying cursor rectangle
#      echo -n "Automatically logging out (type Control-C to
interrupt)"
#      sleep 5
#      logout              # logout after leaving windows system
#      breaksw
#      #
#endsw
      logout              # logout after leaving windows system
```

Archivo: .openwin-init

```
#!/bin/sh
# .openwin-init - OpenWindows initialization script.
# WARNING: This file is automatically generated.
#       Any changes you make here will be lost!
export DISPLAY
#IFS=.
#SETBASEDISPLAY() { BASEDISPLAY=$1; }
#SETBASEDISPLAY ${DISPLAY}
#IFS=
#SETDISPLAYSCREEN() {
#     DISPLAY=${BASEDISPLAY}.$1
#     if winsysck x11 ; then
#          :
#     else
#          echo No display available for screen $1
#          exit 1
#     fi
#     eval `svenv -env`
#}
# Note: toolwait is a utility to control client startup.
#     For more information, see the toolwait(1) man page.
#
# Start clients on screen 0
#
SETDISPLAYSCREEN 0
#
toolwait clock -Wp 826 2 -Ws 92 63 -WP 159 413 +Wi
toolwait cmdtool -Wp 172 168 -Ws 684 428 -WP 0 0 +Wi
```

Archivo: S95xdmint

```
# xdm startup
state=$1
set `who -r`
case $state in
'start')
        if [ -f /usr/openwin/bin/xdm ]; then
                echo "xdm starting."
                /usr/openwin/bin/xdm -config /usr/lib/X11/xdm/xdm-
config
        fi
        ;;
'stop')
        PID=`/usr/bin/ps -ef | grep /usr/openwin/bin/xdm | awk '{
print $2 }'`
        if [ ! -z "$PID" ]; then
                /usr/bin/kill ${PID} 1>/dev/null 2>&1
        fi
        ;;
*)
        echo "Usage: /etc/init.d/xdminit { start | stop }"
        ;;
esac
exit 0
```

Archivo: .Xdefaults

```
window.y: 0
window.x: 0
window.iconic: False
window.height: 114
window.width: 283
icon.x: 0
icon.y: 0
OpenWindows.ScrollbarPlacement: right
OpenWindows.WindowColor: #cccccc
OpenWindows.SelectDisplaysMenu: False
OpenWindows.DragRightDistance: 100
OpenWindows.MultiClickTimeout: 4
OpenWindows.PopupJumpCursor: True
OpenWindows.Beep: always
OpenWindows.WorkspaceColor: #40a0c0
OpenWindows.SetInput: followmouse
OpenWindows.IconLocation: right
Scrollbar.JumpCursor: True
*inputLang: C
*displayLang: C
*basicLocale: C
*timeFormat: C
*numeric: C
```

Archivo: .xinitrc

```
# .xinitrc - OpenWindows startup script.

if [ -f $HOME/.Xdefaults ]; then
   xrdb $HOME/.Xdefaults &              # Load Users X11 resource
database
else
   xrdb $OPENWINHOME/lib/Xdefaults &    # Load Default X11
resource database
fi
$OPENWINHOME/lib/openwin-sys &          # OpenWindows
system initialization

# Install function key "F1" as an Open Look "Help" key
# This precludes its use by applications
# If your applications use F1 for anything else, comment out the
following line
xmodmap -e 'keysym F1 = Help'

eval `svenv -env`                       # SunView binary compatibility
olvwm -3 &                              # OpenLook Virtual Window
Manager (3-D look)
# /apps/Xapps/xsetbg /home/Morgan/master/elle.im8.Z
if [ -x $HOME/.openwin-init ]; then
   $HOME/.openwin-init                  # Custom OpenWindows
tools
else
   $OPENWINHOME/lib/openwin-init  # Default OpenWindows
tools
fi
wait
```

Apéndice B. Instalación física del equipo

Apéndice B. Instalación física del equipo Sun

La instalación física de un equipo Sun es sumamente sencilla, pero antes de realizarla deben de considerarse algunos puntos.

Primero que nada debemos tener cuidado al momento de desempacar el equipo, para lo cual podemos seguir los siguientes pasos:

1. Inspeccionar la caja de embarque antes de abrirla. Si tiene evidencia de daño el empaque, contactar con el distribuidor que le haya vendido dicho equipo para que esté presente al momento de que se abra, y en el caso de que el equipo no se encuentre en óptimas condiciones, poder hacer valer la garantía.

2. Remover el material de protección del equipo (cubiertas de plástico, unicel, etc).

3. Checar el instructivo que acompañe al equipo y revisar que se encuentren todas las partes especificadas en él. Algunos de éstos pueden ser:

- Un dispositivo periférico, *CPU*, monitor, tarjeta, etc. (Para conocer los dispositivos periféricos más comunes dentro de Sun, ver de las figuras B.1 a la B.3).

- Un cable de corriente

- Un cable *SCSI* (Ver figura B.4)

- Un terminador *SCSI* pasivo (Ver figura B.5) o bien un terminador SCSI regulado (Ver figura B.6)

- Cartucho de cinta (incluída sólo en el caso de que el periférico sea una unidad de cinta)

- Tres *caddies* (incluídos sólo en el caso de que el periférico sea un *SunCD*)

Ejemplos de los dispositivos periféricos mas usuales dentro del equipo Sun:

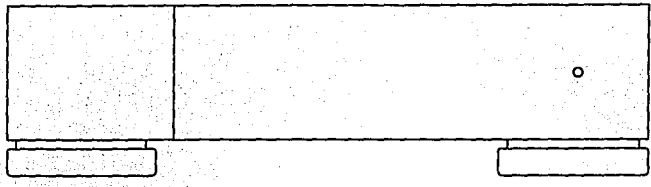


Figura B.1 Disco duro de tipo escritorio

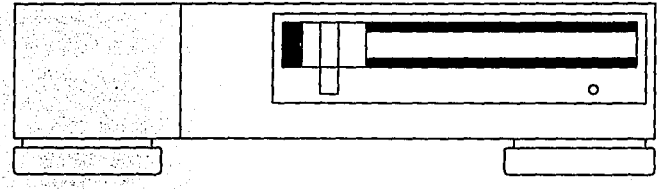


Figura B.2. Unidad de cinta de tipo escritorio

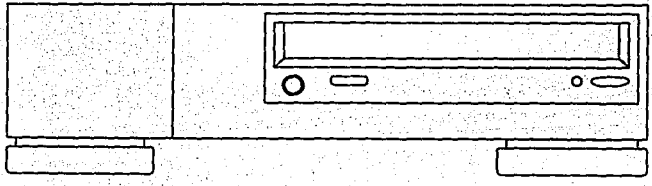


Figura B.3. Unidad de CD de tipo escritorio

Tipos de terminadores, cables y conectores SCSI:

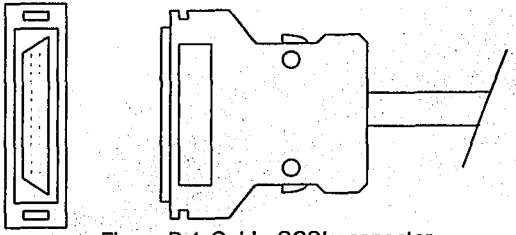


Figura B.4. Cable *SCSI* y conector

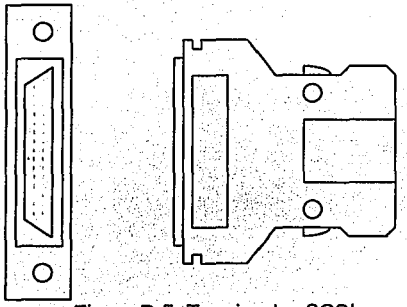


Figura B.5. Terminador *SCSI* y conector

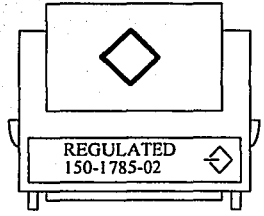


Figura B.6. Terminador *SCSI* regulado

Una vez que se ha desempacado el equipo, deberá procederse a realizar la instalación del equipo, la cual podemos resumir en los siguientes puntos:

1. Antes de comenzar a conectar los dispositivos y cables es necesario, determinar cómo se van a ubicar dentro del área asignada para ellos.

2. Definida la disposición de los periféricos es necesario determinar las direcciones físicas (*TARGETS*) de cada uno. A continuación se muestra una tabla para auxiliarse en la elección de las direcciones.

Tipo de dispositivo	Direcciónes Físicas
Discos duros	0, 1, 2, 3
Discos ópticos	0, 1, 2, 3
Unidades de cinta	4, 5
Unidad de *CDROM*	6

La dirección será asignada a través de switches (Ver figura B.7) o bien de jumpers.

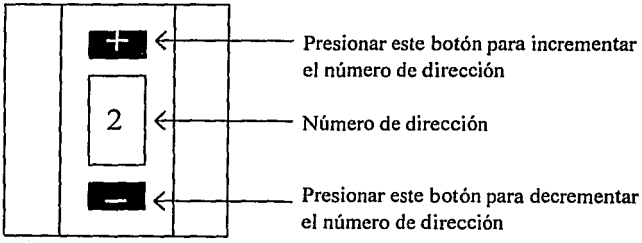


Figura B.7. Switch de direcciones *SCSI*

Nota: Estas direcciones son las que provee el fabricante para la primera tarjeta controladora de dispositivos *SCSI*, pero obviamente estos valores pueden variar.

3. El siguiente paso será la conexión de los diferentes dispositivos a la estación de trabajo, como lo muestra la figura B.8

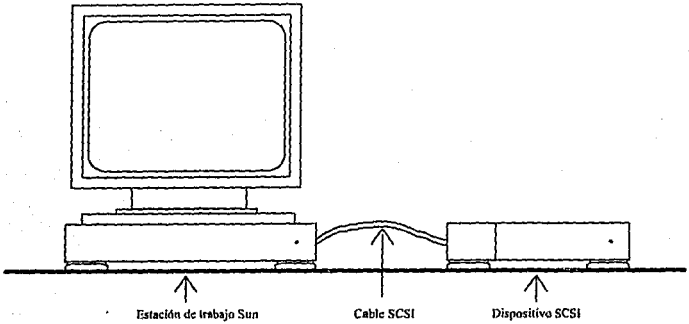


Figura B.8 Conexión de periféricos a la estación de trabajo

Obviamente para realizar dichas conexiones, se debe de saber identificar los diferentes conectores de una estación de trabajo (figura B.9) y de los dispositivos periféricos (ver figura B.10).

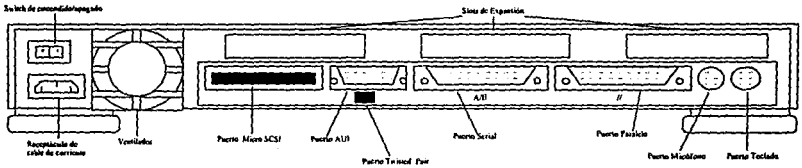
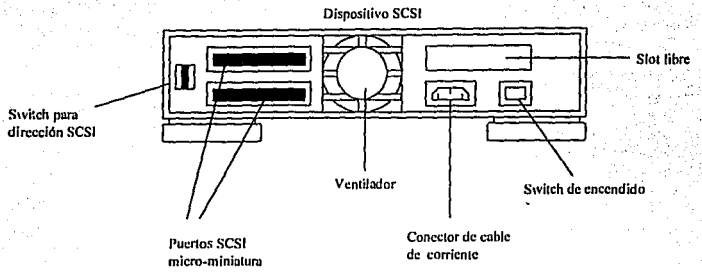


Figura B.9. Vista posterior de una estación de trabajo Sun



B.10. Vista posterior de un dispositivo *SCSI*.

Cuando se tienen plenamente identificadas las diferentes partes que componen la parte trasera de una estación de trabajo y un dispositivo *SCSI* Sun, procederemos a conectar los cables de corriente, el teclado, el monitor y la interfaces *SCSI* de los dispositivos periféricos, de manera de que se forme una cascada de ellos (la salida de un dispositivo. será la entrada de otra y viceversa). Ver figuras B.11 y B.12.

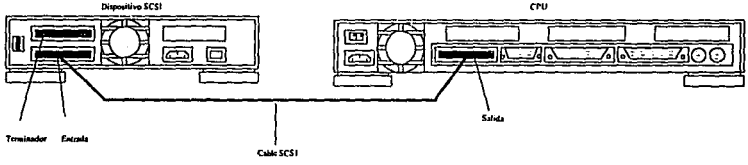


Figura B.11. Conexión de un dispositivo periférico a una estación de trabajo Sun

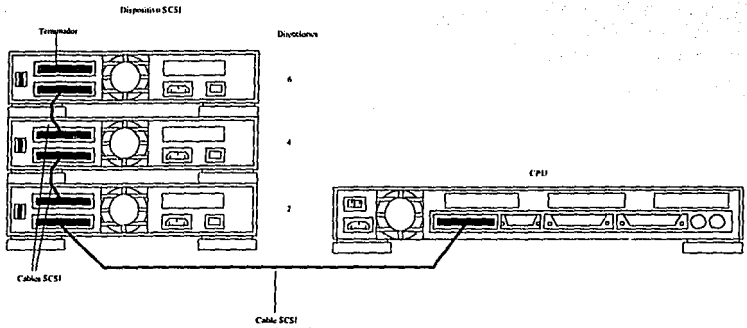


Figura B.12. Conexión de dispositivos periféricos a una estación de trabajo Sun

4. Cuando se han terminado de conectar los dispositivos y los cables de corriente, el siguiente paso es encender el equipo, empezando por las unidades de cinta, los discos, etc., por último el monitor y *CPU.*

5. Una vez que haya encendido el equipo pulsar la combinación de teclas **STOP - A**, con el objeto de que el sistema se encuentre en el modo de diagnóstico de *hardware OBP (Open Boot PROM).*

6. En este caso deberá de aparecer un prompt de **OK**, el cual nos indica que podemos a comenzar ha realizar pruebas de dianóstico de hardware.
7. La primera prueba que se realizará es teclear el comando **banner**, el cual nos indicará la versión del *PROM (Programmable Read Only Memory)*, la dirección *ethernet* de la máquina, el *hostid* y el modelo.

8. Posteriormente otro comando que nos permitirá la prueba de los dispositivos que acabamos de instalar es el **probe-scsi**, el cual tiene por finalidad el reconocimiento de todos los dispositivos *SCSI (Small Computer System Interface)* conectados a la estación de trabajo. En el caso de que reporte un menor número de dispositivos *SCSI* de los que se encuentran instalados, realizar una revisión de las conexiones de

corriente, los cables *SCSI* y de si están encendidos los dispositivos. Si ninguna de estas pruebas dió resultado, checar que alguna de las direcciones asignadas a los dispositivos no se encuentre duplicada. Una vez realizado lo anterior, volver a correr el **probe-scsi**. Si aún así no fué eliminado el problema, consultar con su distribuidor o personal de soporte del fabricante.

9. Posteriormente se conectará el *transceptor* o cable *UTP* al puerto de comunicaciones de la estación de trabajo.

10. Para finalizar se conectarán los cables de corriente y se encenderá el equipo, comenzando por los dispositivos periféricos y por último el monitor y CPU.

Apéndice C. Instalación de un módem bidireccional

Apendice C. Instalación de un módem bidireccional

Antecedentes

La mayoría de los sistemas actuales tiene la necesidad de comunicarse con equipos que se encuentran físicamente distantes, esto lo pueden hacer a través de medios como las microondas, las comunicaciones satelitales, la fibra óptica, etc., pero sin duda alguna uno de lo medios que mayor aceptación han tenido es sin lugar a duda el módem, que tan sólo requiere de una línea telefónica para funcionar. **UNIX** no es la excepción a ello y por eso mismo, a continuación se indica el procedimiento para configurar un módem en modo bidireccional.

Comandos y materiales a utilizar

Un cable serial bajo el estándar *RS-232* (aquí se indica la configuración para conectar un módem Hayes modelo OPTIMA 14400), que tendrá la siguiente configuración:

Computadora		Módem
2	⟷	2
3	⟷	3
4	⟷	4
5	⟷	5
6	⟷	6
7	⟷	7
8	⟷	8
20	⟷	20

Un módem marca Hayes, modelo óptima 14400 (o similar).

Comandos Básicos	Comandos Administración
grep	**ttyadm** **sacadm** **pmadm** **printenv** **setenv** **boot** **reset** **tip**

Procedimiento

1. Determinar qué puerto de la máquina será utilizado, el serial A o el serial B -éste último, sólo puede ser accesado en el caso de que se tenga un *split cable*-.

2. En el caso de que se haya elegido correctamente el puerto serial (en este caso el B), checar a qué velocidad está configurado. Esto se puede hacer a través del *Open Boot PROM* con el siguiente comando y de la salida que se obtenga, checar la correspondiente a **ttyb-ignore-cd** y a **ttyb-rts-dtr-off** por ejemplo:

ok **printenv**

Parameter Name	*Value*	*Default Value*
tpe-link-test?	*true*	*true*
output-device	*screen*	*screen*
input-device	*keyboard*	*keyboard*
sbus-probe-list	*f0123*	*f0123*
keyborad-click?	*false*	*false*
ttyb-rts-dtr-off	*false*	*false*
ttyb-ignore-cd	*true*	*true*
ttya-rts-dtr-off	*false*	*false*
ttya-ignore-cd	*true*	*true*
ttyb-mode	*9600,8,n,1,-*	*9600,8,n,1,-*
ttya-mode	*9600,8,n,1,-*	*9600,8,n,1,-*

More [<space>,<cr>,q] ? q

3. Deshabilitar la señal de detección de carga desde el prompt *PROM.*

ok **setenv** *ttyb-ignore-cd false*

4. Deshabilitar data-terminal-ready (*DTR*) y request-to-send (*RTS*).

ok **setenv** *ttyb-rts-dtr-off true*

5. Después reiniciar el sistema:

ok **boot**

u

ok **reset**

6. Conectar correctamente el módem al puerto de la estación de trabajo.

7. Una vez que ha sido levantado el sistema, teclear el comando ttyadm, con el objeto de desplegar el número de la versión del monitor de puerto actual ttymon:

 # ttyadm *-V*
1

8. Teclear el comando sacadm para adicionar una instancia al monitor de puerto ttymon llamada zsmon.

 # sacadm *-a -p zsmon -t ttymon -c /usr/lib/saf/ttymon -v 1*

donde la opción *-a* sirve para adicionar la instancia, *-p* se refiere a la instancia, *-t* al monitor de puerto y *-c* se refiere al comando relacionado al monitor del puerto.

9. Remover el servicio existente para ttya para que la terminal pueda ser conectada por la versión 2.X de Solaris.

 # pmadm *-r -p zsmon -s ttyb*
donde *-r* es la opción que sirve para remover el servicio, *-p* indica la instancia del puerto y *-s* se refiere al puerto de servicio.

10. Teclear de nueva cuenta el comando pmadm para asociar el monitor del puerto con el nuevo servicio.

 # pmadm *-a -p zsmon -s ttyb -i root -fu -v 1* \
> -m "'ttyadm -b -d /dev/term/b -l conttyH \
> -m ldterm,ttcompat -s /usr/bin/login -S n`"

donde la opción *-a* sirve para adicionar el servicio a la instancia zsmon, *-p* indica el nombre de la instancia, *-s* indica el puerto de servicio, *-fu* crea una entrada en el utmp -archivo para el registro de las sesiones abiertas por los usuarios-, *-v* indica la versión del monitor de puerto, *-m* indica el tipo de servicio a levantar.

11. Asegúrese de que en el archivo **/etc/remote** exista una entrada para el dispositivo serial y que esta tenga la velocidad de transferencia correcta.

 # grep /dev/cua/b /etc/remote
cuab:dv=/dev/cua/b:br#9600
:dv=/dev/cua/b:

12. Editar el archivo **/etc/uucp/Devices** e introducir la siguiente información:

ACU cua/b,M - conttyH hayes

13. Posteriormente encienda el módem e intente entablar una comunicación, por ejemplo:

 # tip *-9600 7213341*

donde 9600 es la velocidad de transmisión y 7213341 es el número telefónico de la máquina remota.

Nota: En este caso lo único que se propone con este apéndice es la forma mediante la cual se puede configurar un módem en modo bidireccional dentro del sistema operativo Solaris 2.X - de manera que pueda hacer o recibir llamadas de otros sistemas y entablar una sesión a través de aplicaciones como lo podría ser *telnet*, más no se pretende ahondar en programas de comunicaciones que permitan la transferencia de archivos como *UUCP*, o algún otro paquete.

Apéndice D. Adicionar cuotas a usuarios

Apéndice D. Adicionar cuotas a usuarios

Antecedentes

En los sistemas multiusuario generalmente la mayoría de los usuarios generan indiscriminadamente archivos de trabajo, siendo este un verdadero problema al momento de hacer respaldos de los diferentes sistemas de archivos, restauración de información perdida o dañada o bien para los mismos usuarios al hacer depuración de su información.

El espacio en disco es siempre un recurso limitado. Una forma de controlar el espacio en disco disponible es a través de configurar y relocalizar espacio dentro de los file systems para los usuarios. Otra forma de controlar el espacio en disco para los *file systems* del tipo (**UFS** - UNIX File Systems) es configurar y administrar las *cuotas* de disco (disk quotas). Las *cuotas* de disco le permiten asignar límites para el almacenamiento en disco duro a cada usuario de un *file system*, controlando de esta manera el máximo número de archivos que cada usuario pueda crear y la máxima cantidad de espacio en disco disponible para cada usuario.

Cada cuota posee dos límites: un *hard limit* y un *soft limit*. Los usuarios diferentes a **root** jamás podrám exceder el *hard limit*. Cuando el *hard limit* es alcanzado, el sistema operativo notifica al usuario y rechaza dar más recursos. Los usuarios pueden exceder el *soft limit* temporalmente, por un período determinado de tiempo. Los usuarios reciben un mensaje de advertencia, y el sistema operativo relocaliza los recursos adicionales. Si el uso en disco excede el soft limit en el siguiente *login*, el mensaje de advertencia en enviado de nueva cuenta. Dependiendo de cómo son implementadas las cuotas, los usuarios que continúan el *soft limit* pueden tener acceso denegado a recursos adicionales hasta que hayan borrado archivos para liberar espacio en disco y éste se encuentre por debajo del *soft limit*.

Si usted implementa un sistema de cuotas, usted debe de decidir cúales *file systems* requieren de quotas. Usualmente, los file systems que contienen información de usuarios. Usted no necesita asignar cuotas a los *file systems* de **/, /usr, /tmp**.

Las utilerías de cuotas le permiten:

- Configurar cuotas para cada usuario en cada *file system.*
- Activar o desactivar el chequeo de cuotas para cada *file system.*
- Reporte de cuotas definidas y uso actual en disco por cada *fs.*
- Sincronizar el uso configurado con el uso actual.

Comandos a utilizar

Comandos Básicos	Comandos Administración
vi	**/etc/rc2.d/S01MOUNTFSYS**
cd	**edquota**
touch	**quotaon**
chmod	**quotaoff**
pwd	**quotacheck**
ls	**repquota**
	df
	su

Procedimiento

1. Convertirse en súper usuario

% su
Password: XXXXXX
#

2. Editar el archivo **/etc/rc2.d/S01MOUNTFSYS**, que es un archivo que le permite al sistema inicializar las cuotas al arrancar el sistema

 # vi */etc/rc2.d/S01MOUNTFSYS*

3. Descomentar las siguientes líneas subrayadas del archivo:

Vista inicial del archivo:

```
#Ident "@(#)MOUNTFSYS 1.9 93/07/07 SMI" /* SVr4.0 1.1.4.1  */
#Mount file systems
cd /
/sbin/mountall -l
# make sure /usr subtree is present by testing for /usr/sbin
if [ ! -d /usr/sbin ]
then
        echo "/usr sub-tree is not present - changing to single \
                user mode"
        /etc/init S
fi
# Check quotas
# Uncomment the following lines to turn on quota checking
# echo 'checking quotas: \c'
# quotacheck -a -p
# echo 'done.'
# quotaon -a
```

de esta forma deberá de lucir el archivo después de la edición:

```
#Ident "@(#)MOUNTFSYS 1.9 93/07/07 SMI" /* SVr4.0 1.1.4.1 */
#Mount file systems
cd /
/sbin/mountall -l
# make sure /usr subtree is present by testing for /usr/sbin
if [ ! -d /usr/sbin ]
then
        echo "/usr sub-tree is not present - changing to \
                single user mode"
        /etc/init S
fi
# Check quotas
# Uncomment the following lines to turn on quota checking
echo 'checking quotas: \c'
quotacheck -a -p
echo 'done.'
quotaon -a
```

4. Así mismo, se deberán de adicionar entradas en el archivo **/etc/vfstab** para activar las cuotas en los file systems que se especifiquen cada vez que ellos sean montados; esto puede hacerse a través del editor que desee, en este caso con el **vi**

 # vi */etc/vfstab*

el contenido del archivo puede lucir de la siguiente forma

```
#device     device     mount     FS    fsck  mount     mount
#to mount  to fsck    point     type  pass  at boot   options
#
/proc       -          /proc     proc  -     no        -
fd          -          /dev/fd   fd    -     no        -
swap        -          /tmp      tmpfs -     yes       -
/dev/dsk/c0t3d0s0  /dev/rdsk/c0t3d0s0  /     ufs   1     no
-
/dev/dsk/c0t3d0s1  -  -  swap  -  no  -
/dev/dsk/c0t3d0s3  /dev/rdsk/c0t3d0s3  /var  ufs   4     no
-
/dev/dsk/c0t3d0s5  /dev/rdsk/c0t3d0s5  /opt  ufs   6     yes   -
/dev/dsk/c0t3d0s6  /dev/rdsk/c0t3d0s6  /usr  ufs   2     no
-
/dev/dsk/c0t3d0s7  /dev/rdsk/c0t3d0s7  /pypsa ufs  5     yes   -
/dev/dsk/c0t0d0s1  -  -  swap  -  no  -
/dev/dsk/c0t0d0s4  /dev/rdsk/c0t0d0s4  /usr/acad  ufs  9
yes  -
/dev/dsk/c0t0d0s5  /dev/rdsk/c0t0d0s5  /iiasa ufs  8     yes   -
/dev/dsk/c0t0d0s6  /dev/rdsk/c0t0d0s6  /sparc20  ufs  10
     yes  -
/dev/fd0a  -  /floppy  pcfs  -  no  -
/dev/dsk/c0t6d0s0  -  /cdrom  hsfs  -  no  ro
/dev/dsk/c0t1d0s6  /dev/rdsk/c0t1d0s6  /iiasa2 ufs - yes -
```

para editarlo será necesario adicionar la palabra rq en el campo correspondiente a opciones de montaje (**mount options**)

```
#device    device    mount     FS    fsck  mount     mount
#to mount  to fsck   point     type  pass  at boot   options
#
/proc      -         /proc     proc  -     no        -
fd         -         /dev/fd   fd    -     no        -
swap       -         /tmp      tmpfs -     yes       -
/dev/dsk/c0t3d0s0  /dev/rdsk/c0t3d0s0  /    ufs   1     no
-
/dev/dsk/c0t3d0s1  -  -  swap  -  no  -
/dev/dsk/c0t3d0s3  /dev/rdsk/c0t3d0s3  /var  ufs   4     no
-
/dev/dsk/c0t3d0s5  /dev/rdsk/c0t3d0s5  /opt  ufs   6     yes   -
/dev/dsk/c0t3d0s6  /dev/rdsk/c0t3d0s6  /usr  ufs   2     no
-
/dev/dsk/c0t3d0s7  /dev/rdsk/c0t3d0s7  /pypsa ufs   5     yes rq
/dev/dsk/c0t0d0s1  -  -  swap  -  no  -
/dev/dsk/c0t0d0s4  /dev/rdsk/c0t0d0s4  /usr/acad  ufs  9
yes  -
/dev/dsk/c0t0d0s5  /dev/rdsk/c0t0d0s5  /iiasa ufs  8     yes   -
/dev/dsk/c0t0d0s6  /dev/rdsk/c0t0d0s6  /sparc20  ufs  10
    yes  -
/dev/fd0a  -  /floppy  pcfs  -  no  -
/dev/dsk/c0t6d0s0  -  /cdrom  hsfs  -  no  ro
/dev/dsk/c0t1d0s6  /dev/rdsk/c0t1d0s6  /iiasa2 ufs - yes -
```

salvar y salir del archivo.

5. Cambiarse al directorio en el cual está montado el *file system* al cual le serán configuradas las cuotas

 # cd */pypsa*

6. Generar un archivo vacío, para que en él se almacene la información de las cuotas

 # touch *quotas*

7. Cambiar los permisos del archivo para que sólo pueda ser leído y escrito únicamente por el usuario **root**

 # chmod *600 quotas*

8. Después para configurar las cuotas por usuario, deberá ser utilizado el comando edquota de la siguiente forma:

 # edquota *sandy*

donde **edquota** es el comando y *sandy* es el nombre del usuario, en seguida aparecerá la siguiente línea:

fs /pypsa blocks (soft = 0, hard = 0) inodes (soft = 0, hard = 0)

en donde el directorio seguido de la palabra *fs* es el directorio de montaje del *file system* al cual le serán asignadas las cuotas; la palabras *blocks* indica el número de bloques de 512 bytes que podrán ser utilizados por el usuario, las palabras *soft* y *hard* indican los límites que tendrán los usuarios; la palabra inodes se refiere básicamente al número de archivos que podrán ser generados por el usuario en ese *file system*, al igual que en *blocks* se poseen los límites *soft* y *hard*.

Para configurar estos valores basta con cambiarlos a través del editor de textos que proporciona el mismo comando **edquota**, que de hecho el el mismo **vi**, por ejemplo:

fs /pypsa blocks (soft = 18000, hard = 20000) inodes (soft = 900, hard = 1000)

salvar y salir. (Para mayores informes sobre el uso del comando **vi**, referirse al manual del sistema operativo **UNIX** o bien a un libro de Fundamentos de **UNIX**).

9. Para activar el chequeo de cuotas usar el siguiente comando:

 # quotaon */pypsa*

donde quotaon es el comando para realizar dicha tarea y */pypsa* es el *file system* que será continuamente checado.

10. Si ya han sido activadas las cuotas, se puede correr periódicamente el comando **quotacheck**, con la finalidad de comparar las cuotas configuradas con los valores actualmente en uso, de la siguiente forma:

 # quotacheck */pypsa*

donde **quotacheck** es el comando que permite realizar dicha comparación y */pypsa* es el *file system* a ser comparado.

11. Si se desea saber en que cantidad están haciendo uso del disco los usuarios, se puede utilizar el siguiente comando:

 # quot */pypsa*
/dev/dsk/c0t3d0s7 (/pypsa):
73298 israel
42652 root
39644 informix
11289 sandy
153 elias
94 captura
50 ricardor
33 raulg
2 francisc
1 gmexico
1 iiasa
#

donde **quot** es el comando y */pypsa* es el *file system* a checar.

12. Si se desea modificar el límite inferior de tiempo en los bloques y los inodos, se puede realizar de la siguiente forma:

 # edquota *-t*

y seguido aparecerá la siguiente línea:

fs /pypsa blocks time limit = 0 (default), files time limit = 0 (default)

en donde */pypsa* es el *file system* sobre el cual se modificarán los límites de tiempo de las cuotas, *blocks time limit* es el límite de tiempo para los bloques y *files time limit* es el límite de tiempo para los archivos. Estos límites como se había mencionado anteriormente sirven para llevar un mejor control en la seguridad de las cuotas. Para modificarlos tan sólo basta con cambiar el valor que tiene cada uno de ellos, salvar el archivo y salir.

Nota: La salida está dada en bloques de 1024 bytes.

Mensajes de error en el caso de que los límites de las cuotas sean excedidos por un usuario:

$ pwd
/pypsa/sandy
$ ls
cuatro isrra local.login sandy tres
dos local.cshrc local.profile sandy2 uno
$ touch *perro*
quota_ufs: Warning: too many files (pid 8627, uid 20000, fs /pypsa)
$ ls
cuatro local.cshrc perro tres
dos local.login sandy uno
isrra local.profile sandy2
$ touch *vampiro gato*

```
$ ls
cuatro      isrra       local.profile  sandy2      vampiro
dos         local.cshrc  perro         tres
gato        local.login  sandy         uno

$ touch a1 a2 a3 a4
quota_ufs: over file hard limit (pid 8633, uid 20000, fs /pypsa)
touch: a4 cannot create
$ ls
a1          dos         local.login    sandy2
a2          gato        local.profile  tres
a3          isrra       perro          uno
cuatro      local.cshrc sandy          vampiro
$
```

Apéndice E. Restaurar el file system de root (/)

Apéndice. Restaurar el file system de root (/)

Antecedentes

Como se ha mencionado anteriormente el file system de root (/) contiene archivos críticos para el sistema, tales como archivos de configuración del sistema, archivos de arranque y bases de datos. En algunas ocasiones si el servidor no es apagado de forma correcta o se llega a dañar alguno de esos archivos el resultado puede ser que el sistema no pueda ser levantado.

En este caso una posible solución podría ser el restaurar la partición de root a partir de un respaldo que se haya efectuado de la misma cuando se encontraba funcionando adecuadamente.

Comandos a utilizar

Comandos Básicos	Comandos Administración
ls	**ufsdump**
rm	**tapes**
cd	**ufsrestore**
	installboot
	fsck
	newfs
	mount
	umount
	reboot
	init
	boot

Procedimiento

1. Respaldar la partición de root. Para respaldar la partición de root, lo más recomendable es reinicializar el sistema a partir del *CDROM* y montar la partición de root en éste, con el objeto de eliminar cualquier tipo de dependencia lógica del sistema con ella; para realizarlo se usarán las siguientes instrucciones:

2 Para reinicializar el sistema a partir del cdrom, se deberá de insertar el *CDROM* del sistema operativo en la unidad de *CD* y después usar:

reboot *cdrom -sw*

el comando reboot tiene por objeto reinicializar el sistema el parámetro **cdrom** le indica al comando el dispositivo que será utilizado para arrancar el sistema y las opciones **-sw** sirven para indicar que el nivel de ejecución será *single-user* y se tendrá la posibilidad de simular escritura en el *cdrom*

o bien

init *0*

para que el nivel de ejecución del sistema pase a 0 y con esto quedará inactivo el sistema operativo

ok **boot** *cdrom -sw*

el prompt **ok** indica que estamos en la modalidad de diagnóstico de hardware y que podemos realizar ciertos procedimientos como cambiar la velocidad de los puertos seriales del equipo, la resolcion de la tarjeta de video, adicionar periféricos, probar la cascada de dispositivos *SCSI*, etc.; entre algunas de esas funciones está la del levantar el sistema con el comando boot, el parámetro cdrom sirve para indicar que éste será el dispositivo de arranque y la opción -sw sirve para indicar que el nivel de ejecución será single-user y se tendrá la posibilidad de simular escritura en el cdrom.

3. Una vez que ha sido levantado el sistema operativo usar el comando **tapes**

 # tapes

éste tiene por objeto que el sistema operativo reconozca las unidades de cinta que se tienen conectadas.

4 Después se procederá a montar la partición correspondiente a root (/) en el CDROM.

 # mount */dev/dsk/c0t3d0s0 /mnt*

en este caso la partición */dev/dsk/c0t3d0s0* corresponde a *root* y *mnt* es un directorio del *CDROM*.

5 Hacer el respaldo con el comando **ufsdump** de la siguiente manera:

 # ufsdump *0uf /dev/rmt/0 /mnt*

para mayor informacion del comando **ufsdump** ver el apéndice de respaldos.

6. Una vez que se ha terminado el respaldo regresar al nivel multiusuario de la siguiente forma:

 # reboot
o
 # init *6*

7. Si ya se tiene el respaldo realizado, entonces se puede proceder a realizar la restauración del file system de root (/), a través del siguiente procedimiento:

8. Levantar el sistema a partir del *CDROM* en modo single-user

ok **boot** *cdrom -sw*

ver paso 2 del presente apéndice.

9. Una vez que ha sido levantado el sistema operativo usar el comando **tapes**

 # tapes

éste tiene por objeto que el sistema operativo reconozca las unidades de cinta que se tienen conectadas.

10. Reinicializar el file system de root (/) con el comando **newfs**

 # newfs */dev/rdsk/c0t3d0s0*

consultar apéndice de comandos para el mantenimiento de discos para mayor información del comando **newfs**.

11. Checar la estructura del file system con el comando **fsck**

 # fsck */dev/rdsk/c0t3d0s0*

para mayor información del comando **fsck** ver apéndice de comandos de mantenimiento de discos.

12. Montar la partición de root en el directorio */mnt* o en el directorio */a*

 # mount */dev/dsk/c0t3d0s0 /mnt*

o

 # mount */dev/dsk/c0t3d0s0 /a*

para mayor información con respecto al comando **mount** ver el capítulo 8 de esta tesis.

13. Cambiarse al directorio que se haya elegido

 # cd */a*

14. Restaurar el respaldo

 # ufsrestore *rvf /dev/rmt/0*

para mayor información del comado **ufsrestore** referirse al apéndice de respaldos.

15. Borrar el archivo *restoresymtable*, ya que ocupa demasiado espacio y ya no sirve de nada

 # rm *restoresymtable*

16. Salirse del directorio actual, hacia el root (**/**) del *CDROM*

 # cd */*

17. Desmontar la partición */a* o */mnt*

 # umount */a*
o
 # umount */mnt*

para mayor información del comando **umount** ver capítulo 8.

18. Correr nuevamente el comando **fsck** a la partición de root

 # fsck */dev/rdsk/c0t3d0s0*

esto con la finalidad de descartar posibles errores físicos del disco duro, en el caso de este dañado, respaldar todas las particiones correspondientes al disco de arranque (con la dirección física 3) con el comando **ufsdump** y formatear el disco duro con el comando **format** y luego volver a generar una por una las particiones, para mayor información consultar el apéndice de modificación de particiones. Si aún con esto no se puede reparar el disco será necesario conseguir uno nuevo y volver a construir los file systems.

19. Pasarse al directorio */usr/lib/fs/ufs*

 # cd */usr/lib/fs/ufs*

20.Correr el comando **installboot**

 # installboot *bootblk /dev/rdsk/c0t3d0s0*

este comando tiene por objeto generar el bloque de arranque que posee la partición de root.

21.Por último reinicializar el sistema

 # reboot

o

 # init *6*

Apéndice F. Instalación de una terminal ASCII.

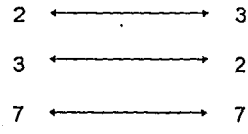
Apéndice F. Instalación de una terminal ASCII.

Antecedentes

La mayoría de los sistemas que operan bajo el sistema operativo **UNIX**, poseen básicamente dos ambientes de trabajo (el ambiente gráfico y el ambiente de texto) y es precisamente por esto que se tiene la necesidad de instalar y configurar terminales *ASCII* dentro de la red.

Comandos y materiales a utilizar

Un *null modem cable* (basado en el estándar *RS-232*), que tendrá la siguiente configuración:



Una terminal *ASCII* (*vt100, vt220, wyse, altos, etc.*)

Comandos Básicos

Comandos Administración

ttyadm
sacadm
pmadm
printenv
setenv

Procedimiento

1. Determinar qué puerto de la máquina será utilizado, el serial A o el serial B -éste último, sólo puede ser accesado en el caso de que se tenga un *split cable*-.

2. En el caso de que se haya elegido correctamente el puerto serial (en este caso el A), checar a qué velocidad está configurado. Esto se puede hacer a través del *Open Boot PROM* con el siguiente comando y de la salida que se obtenga, checar la correspondiente a ttya-mode, por ejemplo:

ok **printenv**

Parameter Name	*Value*	*Default Value*
tpe-link-test?	*true*	*true*
output-device	*screen*	*screen*
input-device	*keyboard*	*keyboard*
sbus-probe-list	*f0123*	*f0123*
keyborad-click?	*false*	*false*
keymap		
ttyb-rts-dtr-off	*false*	*false*
ttyb-ignore-cd	*true*	*true*
ttya-rts-dtr-off	*false*	*false*
ttya-ignore-cd	*true*	*true*
ttyb-mode	*9600,8,n,1,-*	*9600,8,n,1,-*
ttya-mode	*9600,8,n,1,-*	*9600,8,n,1,-*
fcode-debug?	*false*	*false*
diag-file		
diag-device	*net*	*net*
boot device	*disk3*	*disk*
auto-boot?	*true*	*true*

More [<space>,<cr>,q] ? q

3. En el caso de que no fuera la adecuada en cuanto a las características de la terminal, se podría utilizar el siguiente comando:

ok **setenv** *ttya-mode 2400,7,n,1,-*

4. Después reiniciar el sistema:

ok **boot**

5. Una vez que ha sido levantado el sistema, teclear el comando ttyadm, con el objeto de desplegar el número de la versión del monitor de puerto actual **ttymon**:

 # ttyadm *-V*
1#

6. Teclear el comando sacadm para adicionar una instancia al monitor de puerto **ttymon** llamada **zsmon**.

 # sacadm *-a -p zsmon -t ttymon*

donde la opción *-a* sirve para adicionar la instancia, *-p* se refiere a la instancia y *-t* al monitor de puerto.

7. Remover el servicio existente para *ttya* para que la terminal pueda ser conectada por la versión 2.X de Solaris.

 # pmadm *-r -p zsmon -s ttya*

donde *-r* es la opción que sirve para remover el servicio, *-p* indica la instancia del puerto y *-s* se refiere al puerto de servicio.

8. Teclear de nueva cuenta el comando pmadm para asociar el monitor del puerto con el nuevo servicio.

 # pmadm *-a -p zsmon -s ttya -i root -fu -v 1* \
> -m "`ttyadm -I 9600 -d /dev/term/a -T vt100 \
> -i 'Terminal deshabilitada' -s /usr/bin/login -S y`"

donde la opción *-a* sirve para adicionar el servicio a la instancia zsmon, *-p* indica el nombre de la instancia, *-s* indica el puerto de servicio, *-fu* crea una entrada en el utmp -archivo para el registro de las sesiones abiertas por los usuarios-, *-v* indica la versión del monitor de puerto, *-m* indica el tipo de servicio a levantar.

9. Posteriormente encienda la terminal e intente entrar en sesión.

Apéndice G. Configurar un servidor de NFS.

Apéndice G. Configurar un servidor de NFS.

Antecedentes

Como bien es sabida una de las principales limitantes de todos los sistemas actuales, debido a los grandes volúmenes de información es el almacenamiento de ésta en disco, ya que el costo de los mismos en ocasiones es muy elevado y por ende debe de optimizarse al máximo su uso. Es por eso que Sun Microsystems desarrolló una forma para compartir recursos de disco por demás eficiente, que se denomina **NFS** (*Network File System*), esta forma de compartir recursos no sólo puede realizarse entre sistemas con sistema operativo **UNIX**, sino también con *PC's* corriendo **DOS**.

El **NFS** hace irrelevante la actual localización de un file system para el usuario. Usted puede usar **NFS** para permitir que los usuarios puedan ver los archivos que necesiten, en la misma localidad. Asi mismo, permite generar archivos en una localidad y permitir que usuarios de otros servidores puedan accesar a las mismas, esto es muy poderoso, ya que es prácticamente invisible para los usuarios.

Un sistema se convierte en un servidor de **NFS** cuando comparte o exporta sus *file systems* a través de la red. El servidor mantiene una lista de los *file systems* actualmente exportados y de sus restricciones de acceso.

Usted puede desear compartir recursos, como archivos, directorios o dispositivos, desde un sistema en la red con otros sistemas. Por ejemplo, si se tiene alguna aplicación en específico, y se desea compartir con otros servidores, esto se puede realizar mediante **NFS**.

Cuando usted comparte un recurso, usted lo hace disponible para ser montado por un sistema remoto.

Comandos a utilizar

Comandos Básicos	Comandos Administración
more	**share**
vi	**shareall**
ls	**nfs.server**
ps	**showmount**
	rwall
	wall
	reboot
	init

Procedimiento

1. Para poder constituirse como un servidor de **NFS**, primero que nada se debe de verificar de que estén corriendo los procesos correspondientes en la máquina que será destinada para tal fin. Esto se puede verificar a través del comando ps con las opciones *eda* (para mayor información acerca del comando ps consultar algún manual de fundamentos de **UNIX**).

 # ps *-eda*

PID TTY	*TIME COMD*
0 ?	*0:01 sched*
1 ?	*0:03 init*
2 ?	*0:00 pageout*
3 ?	*0:34 fsflush*
505 ?	*0:00 sac*
390 ?	*0:01 rpcbind*
506 console	*0:00 sh*
382 ?	*0:00 in.route*
430 ?	*0:00 syslogd*
417 ?	*0:00 statd*
413 ?	*0:00 automoun*

```
397 ?          0:00 kerbd
392 ?          0:00 keyserv
406 ?          0:01 inetd
419 ?          0:01 lockd
450 ?          0:00 lpsched
458 ?          0:00 lpNet
440 ?          0:00 cron
459 ?          0:00 sendmail
479 ?          0:00 vold
516 console    0:00 xinit
508 ?          0:00 ttymon
517 console    0:17 Xsun
504 ?          0:00 mountd
512 console    0:00 openwin
518 console    0:00 sh
565 pts/2      0:00 ksh
523 console    0:00 fbconsol
530 console    0:00 vkbd
537 console    0:01 olwm
533 console    0:00 ttsessio
534 ?          0:00 rpc.ttdb
538 console    0:00 olwmslav
571 pts/2      0:01 tbmonito
546 console    0:02 cmdtool
548 pts/1      0:00 sh
552 pts/2      0:00 sh
739 pts/4      0:03 ps
573 pts/2      0:00 tbinit
574 pts/2      0:00 tbinit
575 pts/2      0:00 tbinit
576 pts/2      0:00 tbinit
577 pts/2      0:00 tbinit
602 ?          0:08 cmdtool
604 pts/4      0:00 sh
694 ?          0:03 textedit
```

de la información que nos arroje el comando ps, los procesos que nos interesan son el **nfsd** y **mountd**.

Para mayor información ver la siguiente tabla:

		nfsd
	Servidor	
		mountd
Procesos NFS		
		lockd
	Cliente	
		statd

2. En el caso de que no se encuentren corriendo, deberemos de checar la existencia del archivo **/etc/dfs/dfstab**, y en el caso de que exista, que tenga alguna entrada en él. A través del comando more lo podemos hacer.

 # more */etc/dfs/dfstab*

#place share(1M) commands here for automatic execution
#on entering init state 3.
#
#share [F fstype] [-o options] [-d "<text>"] <pathname> [resource]
#.e.g,
#share -F nfs -o rw=engineering -d "home dirs" /export/home2

En este caso el archivo existe, pero no tiene ninguna instrucción para exportar algún *file system*; por ello, la acción a seguir es editar el archivo e incluir en él todos aquellos *file systems* que se requieran exportar (para mayor información acerca de algunos términos utilizados en el presente apéndice, referirse al capítulo 8 de esta tesis).

Esto se puede hacer con el comando **vi**, por ejemplo:

 # vi */etc/dfs/dfstab*
#place share(1M) commands here for automatic execution
#on entering init state 3.
#

#share [F fstype] [-o options] [-d "<text>"] <pathname> [resource]
#.e.g,
#share -F nfs -o rw=engineering -d "home dirs" /export/home2
share -F nfs -o rw,root=pypsa100 /iiasa/iiasa
share -F nfs -o rw=pypsa100:pypsa1 /usr/share/man
#

las dos últimas líneas son las instrucciones que se añadieron para exportar los file systems de */iiasa/iiasa* y de */usr/share/man*. (La sintaxis del comando **share**, que sirve para exportar los file systems, puede verse más ampliamente en el capítulo 8 de la presente tesis).

3. Checar que los nombres de las máquinas a las cuales les serán otorgados los permisos para compartir recursos vía NFS, estén declarados en el archivo **/etc/hosts**, y en caso de que no sea así, editarlo.

 # vi */etc/hosts*
#
 # Internet host table
#
127.0.0.1	***localhost***	***loghost***
192.9.200.60	***sparc20***	
192.9.200.1	***pypsa1***	
192.9.200.20	***pypsa***	
192.9.200.100	***pypsa100***	
192.9.200.50	***inter***	
192.9.200.51	***cadf2***	

4. Una vez que se han editado los archivos **/etc/dfs/dfstab** y **/etc/hosts**, se pueden levantar los procesos de **NFS** de las siguientes formas:

la primera es cambiarse al directorio **/etc/init.d** y correr el script **nfs.server**, con el parámetro *start* (que indica que los procesos relacionados serán inicializados),

 # cd */etc/init.d*

 # ./nfs.server *start*

la otra es reiniciar el server

 # reboot

o

 # init *6*

5. Una vez que ha concluido el paso anterior, de forma automática son exportados los file systems que estaban dados de alta en el archivo /etc/dfs/dfstab. Para comprobar eso usar el comando share sin argumentos.

```
# share
-    /iiasa/iiasa        rw,root=pypsa100    ""
-    /usr/share/man rw=pypsa100              ""
#
```

si al correrse el comando share manda algún tipo de error, o bien no hubo alguna salida con relación a los file systems que se indicaron en el archivo **/etc/dfs/dfstab**. Continuar con el paso 6.

6. Checar nuevamente con el comando **ps** que se encuentren corriendo los procesos antes mencionados.

```
# ps -eda
  PID TTY          TIME COMD
    0 ?            0:01 sched
    1 ?            0:03 init
    2 ?            0:00 pageout
    3 ?            0:34 fsflush
  505 ?            0:00 sac
  390 ?            0:01 rpcbind
  506 console      0:00 sh
  382 ?            0:00 in.route
```

```
430 ?          0:00 syslogd
417 ?          0:00 statd
413 ?          0:00 automoun
397 ?          0:00 kerbd
392 ?          0:00 keyserv
406 ?          0:01 inetd
419 ?          0:01 lockd
450 ?          0:00 lpsched
458 ?          0:00 lpNet
440 ?          0:00 cron
459 ?          0:00 sendmail
479 ?          0:00 vold
516 console    0:00 xinit
508 ?          0:00 ttymon
502 ?          0:00 nfsd
517 console    0:17 Xsun
504 ?          0:00 mountd
512 console    0:00 openwin
518 console    0:00 sh
565 pts/2      0:00 ksh
523 console    0:00 fbconsol
530 console    0:00 vkbd
537 console    0:01 olwm
533 console    0:00 ttsessio
534 ?          0:00 rpc.ttdb
538 console    0:00 olwmslav
571 pts/2      0:01 tbmonito
546 console    0:02 cmdtool
548 pts/1      0:00 sh
550 console    0:01 cmdtool
552 pts/2      0:00 sh
739 pts/4      0:03 ps
573 pts/2      0:00 tbinit
574 pts/2      0:00 tbinit
575 pts/2      0:00 tbinit
576 pts/2      0:00 tbinit
577 pts/2      0:00 tbinit
602 ?          0:08 cmdtool
```

604 pts/4 0:00 sh
694 ? 0:03 textedit
#

Nota: En el caso de que todavía no se hayan levantado los procesos checar que los pasos anteriores hayan sido realizados correctamente.

7. Para exportar manualmente los *file systems*, deben de tenerse corriendo todos los procesos anteriormente descritos, así como los archivos de configuración, de forma manual se podría hacer de la manera siguiente:

#
share *-F nfs -o rw=pypsa100 /opt*
#

8. Si se desearan exportar todos los file systems que se encuentren en el archivo */etc/dfs/dfstab*, puede ejecutarse el siguente comando:

 # shareall

9. Para imposibilitar el acceso a los *file systems* que se desee, se pueden utilizar los comandos **unshare** (para uno) o **unshareall** (para todos), por ejemplo, si se desea dejar indisponible el *file system* de */opt* para la red se puede usar:

 # unshare */opt*
 # share
- ***/iiasa/iiasa rw,root=pypsa100 ""***

o si se quieren eliminar todos los file systems que se encuentran disponibles actualmente:

 # unshareall
 # share
#

10. Como todos los recursos que se ponen a disposición de una red son críticos, ya que mucha personas pueden estar accesando a éstos de forma simultánea, se tienen que tomar ciertas medidas de seguridad. Por ejemplo:

el comando **showmount** muestra que máquinas se encuentran accesando los *file systems* vía **NFS**;

 # showmount
pypsa
#

sin opciones indica los servidores que hacen uso del servicio de **NFS**

 # showmount *-a*
pypsa100:/usr/informix
#

con la opción a indica que *file systems* están siendo accesados y por qué máquinas. Una vez que se sabe quienes usan los recuros de **NFS**, se les puede avisar a través de un **rwall**, por ejemplo.
 # rwall *pypsa pypsa100 pypsa1*
"Las particiones de /opt y /usr/share/man del servidor sparc20 serán deshabilitadas en 10 minutos"
<ctrl-d>
#

Apéndice H. Optimizar el kernel

Apéndice H. Optimizar el kernel

Antecedentes

Tener un buen performance de una computadora o red es una parte importante de la administracion del sistema. Este capítulo es una vista de algunos de los factores que contribuyen a mantener y administrar el performance de los sistemas de cómputo a su cuidado.

El performance de una computadora depende en cómo el sistema utiliza y localiza sus recursos. Un aspecto importante en el monitoreo del performance de su sistema es que usted conozca cuándo está funcionando en condiciones óptimas y forme una base regular en función a la observación periódica de éste, para que de esta forma usted pueda determinar cuando se encuentra trabajando anormalmente. Usted podrá tener una buena idea de qúe esperar, y ser capaz de reconocer un problema cuando este ocurra.

El performance de una red depende no solamente de los sistemas individuales que conforman la red, sino además de las interrelaciones entre ellos. Cualquier miembro de la red , o de los componentes físicos de la red por sí mismos, los cables y conectores, pueden tener impacto en el performance de la red como si fuera un hoyo. Descubrir y solucionar problemas puede en algunas ocasiones involucrar un gran reto de trabajo minucioso.

Los recursos del sistema incluyen:

Unidad Central de Proceso (*CPU*) - El *CPU* procesa las instrucciones - encontrando las instrucciones en la memoria y ejecutándolas.

Dispositivos de entrada/salida (*I/O*) - Estos dispositivos transfieren informacion dentro o fuera de la computadora. Un dispositivo puede ser un teclado, una terminal, un disco, una impresora, etc.

Memoria - La memoria puede ser designada como primaria o secundaria. La memoria primaria es utilizada para tener el programa y dato que actualmente está siendo ejecutado, la memoria secundaria esta disponible en discos o algún otro medio de almacenamiento (cintas, discos ópticos, diskettes, etc.). La memoria existe como una memoria física dentro del kernel, y memoria secundaria accesible en disco. El performance se ve disminuído cuando los programas que se están ejecutando en el sistema requieren de más memoria de la que está disponible físicamente, ya sea porque hace falta realmente adicionar más memoria o tan sólo reaprovechar la misma, a través de la configuración del kernel.

Comandos a utilizar

Comandos Básicos	Comandos Administración
cd	**reboot**
vi	**init**
cp	**boot**

Procedimiento

1. Si grandes cantidades de memoria están siendo monopolizadas por el *kernel*, usted puede desear relocalizar recursos de *buffers* de datos. Si un gran número de *buffers* han sido configurados, trate de decrementar el número de éstos (almacenados en el parámetro modificable, **p_nbuf,** del archivo **/etc/system**). De esta forma es posible que se regrese espacio ocupado de la memoria por algunos *buffers* con lo cual se podría resolver el problema del *swapeo* al dejar más espacio para los programas de los usuarios.

2. Muchos parámetros básicos dentro del *kernel* son calculados a partir del valor del parámetro, **maxusers**. Cambiandar el valor de **maxusers** es la forma más segura y usada para optimizar un sistema.

Esta variable del *kernel* contiene el número aproximado de usuarios activos dentro del sistema. Por default está asignado un valor de 8 usuarios en el *kernel* **GENERIC**, pero puede ser reconfigurado para reflejar la carga actual dentro del server. Usted puede definir un usuario por cada usuario de tiempo compartido en el sistema, uno por cada ventana usualmente utilizada, y uno por cada *diskless client* configurado dentro del sistema.

Si la cantidad de memoria es suficiente, incrementar el valor de **maxusers** mejora los rangos de uso del *caché*, con lo cual se mejora el performance de **NFS**. Una fórmula para configurar servidores con 32 *Mbytes* de *RAM* es: Con cuatro o menos discos y 10 ó menos usuarios concurrentes en el sistema, configurar la variable de maxusers en **64**. Si existen más de cuatro discos y más de 10 usuarios concurrentes dentro del sistema, incrementar la variable de **maxusers** hasta 128.

Un número de parámetros del *kernel* configuran sus valores por default en base al valor de **maxusers**. Por ejemplo, **maxusers** afecta los siguientes parámetros:

ncallout El tamaño de la tabla **callout**.

ufs_ninode El tamaño de la tabla de *inodos*.

ncsize El tamaño del directorio llamado **lookup cache**.

max_nprocs El tamaño de la tabla de procesos.

ndquot El número de estructuras de *cuotas* en disco.

maxuprc El número de procesos de usuario.

La siguiente tabla lista los valores por *default* de los parámetros del *kernel* afectados por el valor asignado a **maxusers**.

Tabla del kernel	Variable	Valor configurado
Callout	**ncallout**	16+**max_nprocs**
Inode	**ufs_ninode**	**max_nprocs**+16+**maxusers**+64
Name cache	**ncsizeb**	**max_nprocs**+16+**maxusers**+64
Process	**max_nprocs**	10+16*maxusers
Quota table	**ndquot**	(**maxusers***NMOUNT)/4+**max_nprocs**
User process	**maxuprc**	**max_nprocs**-5

3. Una vez que se tiene en cuenta lo anterior, se puede hacer un ejemplo para configurar algunas variables del kernel, en función a un número de 8 usuarios (físicos) dentro del sistema.

Como se mencionó anteriormente un número aceptable para la variable de **maxusers** será de 64, y en base a ella se harán los cálculos correspondientes a *callout*, *inode*, *name caché, process, quota table* y *user process*, de la siguiente manera:

maxusers = 64

max_nprocs = 10 + 16 * **maxusers** = 10 + 16 * 64 = 1034

ncallout = 16+**max_nprocs** = 16 + 1034 = 1050

ufs_ninode = **max_nprocs**+16+**maxusers**+64 = 1034 + 16 + 64 + 64 = 1178

ncsizeb = **max_nprocs**+16+**maxusers**+64 = 1034 + 16 + 64 + 64 = 1178

ndquot = (**maxusers***NMOUNT)/4+**max_nprocs** = (64*10)/4+1034 = 1194

Nota: NMOUNT se refiere al número de file systems montados en el sistema, y para el caso de este ejercicio se consideraron 10.

maxuprc = **max_nprocs** - 5 = 1034 - 5 = 1029

4. Una vez que se han calculado los valores para las variables del *kernel*, éstos deberán ser grabados en el archivo **/etc/system**, pero antes de eso se deberá hacer una copia de respaldo del mismo.

cp */etc/system /etc/system.orig*

vi */etc/system*

```
...
*ident "@(#)system 1.15 92/11/14 SMI" /* SVR4 1.5 */
*
* SYSTEM SPECIFICATION FILE
*

* moddir:
*
*Set the search path for modules.  This has a format similar to the
*csh path variable.  If the module isn't found in the first directory
*it tries the second and so on.  The default is /kernel /usr/kernel
*
*       Example:
*               moddir: /kernel /usr/kernel /other/modules

* root device and root filesystem configuration:
*
*The following may be used to override the defaults provided by
*the boot program:
*
*       rootfs:         Set the filesystem type of the root.
*
```

```
*      rootdev:    Set the root device.  This should be a fully
*                  expanded physical pathname.  The default is the
*                  physical pathname of the device where the boot
*                  program resides.  The physical pathname is
*                  highly platform and configuration dependent.
*
*      Example:
*              rootfs:ufs
*              rootdev:/sbus@1,f8000000/esp@0,800000/sd@3,0:a
*
*(Swap device configuration should be specified in /etc/vfstab.)

* exclude:
*
*Modules appearing in the moddir path which are NOT to be
*loaded, even if referenced. Note that `exclude' accepts either a
*module name,or a filename which includes the directory.
*
*Examples:
*              exclude: win
*              exclude: sys/shmsys

* forceload:
*
*Cause these modules to be loaded at boot time,
*(just before mounting the root filesystem) rather than at first
*reference. Note that forceload expects a filename which includes
*the directory. Also note that loading a module does not
* necessarily imply that it will be installed.
*
*      Example:
*              forceload: drv/foo

* set:
*
*Set an integer variable in the kernel or a module to a new value.
*This facility should be used with caution.  See system(4).
*
*      Examples:

*      To set variables in 'unix':
```

```
*
*           set nautopush=32
*           set maxusers=40
*
*To set a variable named 'debug' in the module named 'test_module'
*
*           set test_module:debug = 0x13
set maxusers = 64
set max_nprocs=1034
set ncallout= 1050
set ufs_ninode=1178
set ncsizeb=1178
set ndquot=1194
set maxuprc=1029
```

5. Despúes de que se ha modificado el archivo y se ha hecho la copia de respaldo. Bastará con reinicializar el sistema con los comandos, **reboot** o **init** en combinación con **boot**.

 # reboot -- *-r*

la opción *-r* indica que el *kernel* será reconfigurado,

 # init *0*
...
ok **boot** *-r*

Nota: Si por algún motivo el sistema no puede ser inicializado, puede correrse el boot interactivo, de la siguiente forma:

ok **boot** *-a*

y cuando se llegue a la pregunta de cuál es el archivo de configuración para el kernel se le indicará el nombre del respaldo, en nuestro caso **/etc/system.orig**.

Apéndice I. Cambiar el tamaño de una partición

Apéndice I. Cambiar el tamaño de una partición

Antecedentes

Uno de los problemas más frecuentes dentro de los sistemas de cómputo es la falta de espacio para almacenamiento en disco duro, y es aqui donde se comienzan a plantear una serie de expectativas para resolver dicho problema.

Una de ellas sería adquirir más dispositivos (discos duros) en el caso de que se tengan los recursos económicos suficientes, otra sería respaldar la información en algún otro medio de almacenamiento como unidades de cinta, discos ópticos, etc., y borrar la información del disco para liberar espacio; hasta el momento las dos opciones anteriores son bastante viables para el caso de que la información contenida en los mismos no se necesite accesar constantemente y el número de dispositivos conectados a la estación de trabajo no haya sido saturado, pero si nos encontramos en el supuesto de que se desee consultar de manera frecuente dicha información y ya no se pueden añadir más dispositivos periféricos al sistema, una de las opciones más y que nos permitirán optimizar los recursos del sistema es reparticionar el disco o discos - o en otras palabras cambiar el tamaño de las particiones -.

Para esto último se proporcionan las siguientes instrucciones para llevarlo a cabo sin ningún problema.

Comandos a utilizar

Comandos Básicos	Comandos Administración
rm	**ufsrestore**

ls
cd

Comandos Básicos

installboot
fsck

Comandos Administración

newfs
mount
umount
reboot
init
boot
ufsdump
tapes

Procedimiento

1. Determinar el tamaño de las particiones del(os) disco(s) a través del comando df:

```
# df -k
Filesystem            kbytes    used   avail capacity  Mounted on
/dev/dsk/c0t3d0s0      43423   26805   12278    69%    /
/dev/dsk/c0t3d0s6     226759  197134    6955    97%    /usr
/proc                      0       0       0     0%    /proc
fd                         0       0       0     0%    /dev/fd
/dev/dsk/c0t3d0s3      19183    7561    9712    44%    /var
swap                  122876  122652     224    99%    /tmp
/dev/dsk/c0t3d0s5     143641   94924   34357    73%    /opt
/dev/dsk/c0t3d0s7     480815  151593  281142    35%    /pypsa
/dev/dsk/c0t0d0s4     288391  122754  136807    47%    /usr/acad
/dev/dsk/c0t0d0s5     288391  187153   72408    72%    /iiasa
/dev/dsk/c0t0d0s6     101007    3084   87823     3%    /sparc20
/dev/dsk/c0t1d0s6     963662  595967  271335    69%    /iiasa2
/dev/dsk/c0t4d0s2     537784  365387  118627    75%    /usr2
#
```

la opción **-k** en el comando df sirve para que la salida del mismo sea presentada en kilobytes (Kb). En este caso la partición */opt* destinada

a la instalación de aplicaciones está al 73% de su capacidad (con 34357 Kb libres) y un paquete de aplicación que desea instalarse requiere por lo menos de 40000 Kb libres. En este caso podemos ver que la partición de */var* está al 44% de su capacidad y en general el sistema no tiene una gran volumen de correo electrónico, ni de impresiones y colas de espera en general, por lo que se puede tomar de esta partición unos 6000 Kb para añadirlos a la partición de */opt*.

2. Cambiarse a modo single user, con cualquiera de los siguientes comandos:

 # init 1
o

 # shutdown *-i1 -g0*

o
 # reboot -- *-s*

3. Antes de comenzar a reparticionar el disco, se deberá de hacer un respaldo en cada uno de los file systems a modificar.

#ufsdump *0uf /dev/rmt/0n /opt*
#ufsdump *0uf /dev/rmt/0 /var*

Nota: Se recomienda utilizar el comando **ufsdump** para respaldar la información, ya que esta diseñado especialmente para el respaldo de file systems (para mayor información acerca de su uso ver recetario de respaldos).

4. Luego deberá ser invocada la utilería **format** para comenzar con la repartición:

 # format
Searching for disks...done

AVAILABLE DISK SELECTIONS:
0. c0t1d0 <SUN1.05 cyl 2036 alt 2 hd 14 sec 72>
/iommu@f,e0000000/sbus@f,e0001000/espdma@f,400000/esp@f,800000/sd@1,0
1. c0t3d0 <SUN1.05 cyl 2036 alt 2 hd 14 sec 72>
/iommu@f,e0000000/sbus@f,e0001000/espdma@f,400000/esp@f,800000/sd@3,0
2. c0t4d0 <HP C1716T cyl 3550 alt 2 hd 9 sec 36>
/iommu@f,e0000000/sbus@f,e0001000/espdma@f,400000/esp@f,800000/sd@4,0
3. chunk <SUN1.05 cyl 2036 alt 2 hd 14 sec 72>
/iommu@f,e0000000/sbus@f,e0001000/espdma@f,400000/esp@f,800000/sd@0,0

Specify disk (enter its number): 1

En la primera pantalla se muestran todos los discos duros disponibles dentro del sistema, y se pide que se indique cuál será usado y después de que hayamos indicado cúal; en el caso de que este formateado y en uso nos mandará el siguiente mensaje:

selecting c0t3d0
[disk formatted]
Warning: Current Disk has mounted partitions.

La siguiente pantalla tiene por objeto mostrar el menu de opciones que están disponibles en el format

FORMAT MENU:

disk - select a disk
type - select (define) a disk type
partition - select (define) a partition table
current - describe the current disk
format - format and analyze the disk
repair - repair a defective sector
label - write label to the disk
analyze - surface analysis
defect - defect list management
backup - search for backup labels
verify - read and display labels
save - save new disk/partition definitions
inquiry - show vendor, product and revision
volname - set 8-character volume name
quit

format> partition

De todas ellas la que nos interesa es la de **partition**, ya que en base es ésta se pueden modificar los tamaños de las particiones. Una vez que la elegimos, aparacerá el menú correspondiente a las particiones.

PARTITION MENU:

0 - change `0' partition
1 - change `1' partition
2 - change `2' partition
3 - change `3' partition
4 - change `4' partition
5 - change `5' partition
6 - change `6' partition
7 - change `7' partition
select - select a predefined table
modify - modify a predefined partition table
name - name the current table
print - display the current table
label - write partition map and label to the disk
quit

partition> print

la opción de print tiene por objeto desplegar la tabla de particiones

Current partition table (original):

Part	*Tag*	*Flag*	*Cylinders*	*Size*	*Blocks*
0	*root*	*wm*	*0 - 91*	*45.28MB*	*(92/0/0)*
1	*swap*	*wu*	*92 - 193*	*50.20MB*	*(102/0/0)*
2	*backup*	*wm*	*0 - 2035*	*1002.09MB*	*(2036/0/0)*
3	*var*	*wm*	*194 - 234*	*20.18MB*	*(41/0/0)*
4	*unassigned*	*wm*	*0 - 0*	*0MB*	*(0/0/0)*
5	*unassigneg*	*wm*	*235 - 540*	*150.61MB*	*(306/0/0)*
6	*usr*	*wm*	*541 - 1019*	*235.76MB*	*(479/0/0)*
7	*home*	*wm*	*1020 - 2035*	*500.06MB*	*(1016/0/0)*

Después de que se mostró la tabla de particiones, se puede elegir alguna con sólo escribir el número que le corresponde dentro de la tabla, en este caso es el 3.

partition> 3

Después se muestran los valores correspondientes únicamente a esa partición y se piden los nuevos valores:

Part	*Tag*	*Flag*	*Cylinders*	*Size*	*Blocks*
3	*var*	*wm*	*194 - 234*	*20.18MB*	*(41/0/0)*

Enter partition id tag[var]:

introducir nombre de identificador para la partición por defalut el valor se encuentra entre corchetes,

Enter partition permission flags[wm]:

introducir los permisos, que pueden ser *wm* (writeable-mountable) o bien *wu* (writeable-unmountable),

Enter new starting cyl[194]:

introducir el cilindro en el cual comenzará la partición, en este caso se conservó, pero si la partición que vamos a expander estuviera localizada en los cilindros previos de ésta, entonces el cilindro de inicio tendría que recorrerse tantos cilindros como fuera necesario,

Enter partition size[41328b, 41c, 20.18mb]: 14mb

después se puede introducir el tamaño de la partición, y este puede estar dado en bloques (512 bytes), cilindros (512 Kbytes) o en megabytes. En este caso se elegieron megabytes, ya que la cantidad de espacio que le deseamos quitar a la partición eran al rededor de 6 megabytes,

partition> print
Current partition table (unnamed):

Part	Tag	Flag	Cylinders	Size	Blocks
0	root	wm	0 - 91	45.28MB	(92/0/0)
1	swap	wu	92 - 193	50.20MB	(102/0/0)
2	backup	wm	0 - 2035	1002.09MB	(2036/0/0)
3	varwm	194 - 222		14.27MB	(29/0/0)
4	unassignedwm		0 - 0	0MB	(0/0/0)
5	-	wm	235 - 540	150.61MB	(306/0/0)
6	usr	wm	541 - 1019	235.76MB	(479/0/0)
7	home	wm	1020 - 2035	500.06MB	(1016/0/0)

una vez que se modificó el tamaño de la partción se puede volver al listar la tabla de particiones con el objeto de verificar que los cambios realizados son correctos,

partition> 5

Part	Tag	Flag	Cylinders	Size	Blocks
5	-	wm	235 - 540	150.61MB	(306/0/0)

luego, se selecciona la partición número 5 y se modifican sus valores,

Enter partition id tag[unassigned]:

la etiqueta de identificación de la partición no está asignada, pero no es necesario ponerla, por lo cual podemos pulsar *ENTER* para continuar,

Enter partition permission flags[wm]:

los permisos deben conservarse, ya que sólo para el área de *swap* se usan los de *wu*,

Enter new starting cyl[235]: 223

el cilindro de comienzo será modificado, ya que se tomarán los últimos cilindros que correspondían a la partición 3, para no equivocarse basta

con sumarle uno al cilindro final de intervalo correspondiente a la partición 3, es decir del intervalo *194 - 222* se deberá tomar 222 y a éste sumarle *1*, (222 + 1 = **223**)

Enter partition size[308448b, 306c, 150.61mb]: 318c

en este caso el nuevo tamaño de la partición será más fácil indicarlo en cilindros para que sea exacto, ya que sólo bastará con hacer una diferencia aritmética entre el cilindro final del intervalo correspondiente a la partición 5 original menos el nuevo cilindro de inicio + 1, es decir, 540 (cilindro final partición original) - 223 (nuevo cilindro inicial) + 1 = 318 (número de cilindros de la nueva partición 5),

partition> print
Current partition table (unnamed):

Part	*Tag*	*Flag*	*Cylinders*	*Size*	*Blocks*
0	*root*	*wm*	*0 - 91*	*45.28MB*	*(92/0/0)*
1	*swap*	*wu*	*92 - 193*	*50.20MB*	*(102/0/0)*
2	*backup*	*wm*	*0 - 2035*	*1002.09MB*	*(2036/0/0)*
3	*varwm*	*194 - 222*		*14.27MB*	*(29/0/0)*
4	*unassigned*	*wm*	*0 - 0*	*0MB*	*(0/0/0)*
5	*-*	*wm*	*223 - 540*	*156.52MB*	*(306/0/0)*
6	*usr*	*wm*	*541 - 1019*	*235.76MB*	*(479/0/0)*
7	*home*	*wm*	*1020 - 2035*	*500.06MB*	*(1016/0/0)*

posteriormente se verifican todos los intervalos de las diferentes particiones, con el objeto de que no existan cilindros sin utilizar,

partition> label
Ready to label disk, continue? yes
partition> quit

por último, se etiqueta el disco con el comando **label** y se sale del menú con la opción **quit**,

FORMAT MENU:

disk - select a disk
type - select (define) a disk type
partition - select (define) a partition table
current - describe the current disk
format - format and analyze the disk
repair - repair a defective sector
label - write label to the disk
analyze - surface analysis
defect - defect list management
backup - search for backup labels
verify - read and display labels
save - save new disk/partition definitions
inquiry - show vendor, product and revision
volname - set 8-character volume name
quit

format> quit
#

para salir de la utilería del format escribir **quit** y pulsar ENTER.

Hasta este momento ya se tienen redefinidas las áreas de las particiones modificadas.

5. Para generar la estructura de los file systems en base a las nuevas dimensiones deberá correrse el comando newfs sobre cada una de ellas de la siguiente manera:

newfs */dev/rdsk/c0t3d0s3*
/dev/rdsk/c0t3d0s5: 308448 sectors in 29 cylinders of 11 tracks, 72 sectors
14.7MB in 20 cyl groups (16 c/g, 7.88MB/g, 3776 i/g)
super-block backups (for fsck -F ufs -o b=#) at:
32, 16240, 32448, 48656, 64864, 81072, 97280, 113488, 129696,
145904, 162112, 178320, 194528, 210736, 226944, 243152, 258080, 274288,
290496, 306704,

newfs */dev/rdsk/c0t3d0s5*
/dev/rdsk/c0t3d0s5: 308448 sectors in 318 cylinders of 14 tracks, 72 sectors
156.5MB in 20 cyl groups (16 c/g, 7.88MB/g, 3776 i/g)
super-block backups (for fsck -F ufs -o b=#) at:
32, 16240, 32448, 48656, 64864, 81072, 97280, 113488, 129696,
145904, 162112, 178320, 194528, 210736, 226944, 243152, 258080, 274288,
290496, 306704
#

Para mayor información acerca del comando **newfs** referirse al manual de referencia de UNIX, o bién algún manual de administración de UNIX.

6. Cuando se ha terminado de generar la estructura de los nuevos file systems, lo que falta es restaurar la información correspondiente a cada uno de ellos, a través del siguiente procedimiento:

mount */dev/dsk/c0t3d0s5 /opt*
cd */opt*
ufsrestore *rvf /dev/rmt/0n*
rm *restoresymtable*
mount */dev/dsk/c0t3d0s3 /var*
cd */var*
ufsrestore *rvf /dev/rmt/0*
rm *restoresymtable*

7. Por último, para verificar que todo el trabajo que realizamos a redituado resultados podemos comprobar el espacion en disco nuevamente con el comando df.

df -*k*

Filesystem	*kbytes*	*used*	*avail*	*capacity*	*Mounted on*
/dev/dsk/c0t3d0s0	*43423*	*26805*	*12278*	*69%*	*/*
/dev/dsk/c0t3d0s6	*226759*	*197134*	*6955*	*97%*	*/usr*
/proc	*0*	*0*	*0*	*0%*	*/proc*
fd	*0*	*0*	*0*	*0%*	*/dev/fd*
/dev/dsk/c0t3d0s3	*19183*	*7561*	*9712*	*82%*	*/var*
swap	*122876*	*122652*	*224*	*99%*	*/tmp*
/dev/dsk/c0t3d0s5	*143641*	*94924*	*34357*	*61%*	*/opt*
/dev/dsk/c0t3d0s7	*480815*	*151593*	*281142*	*35%*	*/pypsa*
/dev/dsk/c0t0d0s4	*288391*	*122754*	*136807*	*47%*	*/usr/acad*
/dev/dsk/c0t0d0s5	*288391*	*187153*	*72408*	*72%*	*/iiasa*
/dev/dsk/c0t0d0s6	*101007*	*3084*	*87823*	*3%*	*/sparc20*
/dev/dsk/c0t1d0s6	*963662*	*595967*	*271335*	*69%*	*/iiasa2*
/dev/dsk/c0t4d0s2	*537784*	*365387*	*118627*	*75%*	*/usr2*

#

Nota: los tamaños de las particiones pueden variar, en función al valor que se tenga de reserva para cada file system (**minfree**).

Apéndice J. Respaldos

Apéndice J. Respaldos

Antecedentes

La regla básica es, "¿Qué tanta información estoy dispuesto a perder?"

Diariamente, sus usuarios están creando, modificando y borrando archivos. Como el administrador, usted tiene que asumir que cada archivo es de gran importancia. Si un archivo contiene datos valiosos, la pérdida puede ser provocar un efecto devastador en caunto al nivel competitivo de su empresa en el mercado.

Cuando los usuarios borran un archivo, por accidente, éste se pierde. Una mala edición puede mutilar el contenido de un archivo. Un comando usado de manera incauta puede borrar por completo un directorio.

Una falla de energía puede dañar su hardware. Un incendio puede destruir todos sus archivos. Estas y otras muchas causas comúnmente son el origen de la pérdida de información en las empresas.

Cuando sus usuarios necesitan restaurar un archivo, usted puede darle a éste la versión más reciente de ese archivo mediante el último respaldo que se hubiera efectuado.

Comandos a utilizar

Comandos Básicos	Comandos Administración
ls	**cpio**
cd	**dd**
pwd	**tar**
rm	**ufsdump**
	ufsrestore
	rsh
	newfs
	mount
	umount
	wall
	rwall
	init
	shutdown
	fsck

ufsdump y ufsrestore

Antecedentes

El comando **ufsdump** es usado para respaldar un filesystem por completo. Esto permite que usted respalde todos los archivos que estén dentro del filesystem, o sólo los archivos modificados después de una fecha específica, o de un conjunto de específicos archivos o directorios. **ufsdump** soporta respaldos de volumen múltiple; un filesystem que es demasiado largo para caber en sólo un volumen, o cinta, puede ser respaldado en dos o más cintas.

Para usar el dispositivo de cinta de un hosts remoto de cinta, usted puede usar el comando **ufsdump** y al momento de indicar el dispositivo se antepone el nombre de la máquina que posee dicho medio seguido de dos puntos (:), que es tan sólo una variante del **ufsdump**.

El comando **ufsrestore** le permite a usted extraer todos los archivos creados mediante el comando **ufsdump**. Usted puede extraer los archivos y directorios, un conjunto especificado, o usted puede elegir el comando con al opción interactiva.

Procedimiento

El comando **ufsdump** es usado para respaldar un filesystem completo.

ufsdump *[opciones [argumentos]] filesystem*

opciones del **ufsdump**:

0 - 9 Opción para indicar el nivel del ufsdump. El nivel 0 es el más bajo nivel (llamado el nivel epoch o full ufsdump), y el nivel 9 es el más alto nivel.

f Especifica el dispositivo hacia el cual se desea que sean escritos los archivos. Este requiere como argumento el nombre del dispositivo. El dispositivo por default es **/dev/rmt8.**

u Actualiza el registro de **ufsdump (/etc/dumpdates)** con la fecha y nivel del ufsdump de este filesystem.

c El **ufsdump** se realiza hacia un cartucho de cinta.

b Elegir el factor de bloqueaje. El default es 20 bloques para cintas de 1/2 pulgada a 1600bpi, 64 para cintas de 1/2 pulgada a 6250bpi y 126 para cintas de 1/4 de pulgada.

Nota: El factor de bloqueaje es el número de bloques de cinta (512 bytes) antes de insertar una **etiqueta inter bloque.**

s Indica la longitud de la cinta. El default es 2300 pies para cintas de 1/2 pulgada, 425 pies para cintas de 1/4 de pulgada de 60MB, 600 pies para cintas de 1/4 de pulgada de 150MB, y 6000 pies para cintas de 8mm de 2.3GB.

d Densidad de la cinta. El default es 1600-bpi para cintas de 1/2 pulgada, y 54,000-bpi para cintas de 8mm de 2.3GB. Es necesario especificar la densidad para los dispositivos de cinta de 6250-bpi.

El comando **ufsrestore** extrae archivos de un respaldo creado con **ufsdump**. Su sintaxis es la siguiente:

ufsrestore *opciones [modificadores] [nombre de archivo...]*

opciones del **ufsrestore**:

t Listar la tabla de contenidos de un respaldo

x Restaurar sólo los archivos nombrados en la línea de comando

r Restaurar completamente el respaldo

i Realizar un ufsrestore interactivo

Algunos modificadores del **ufsrestore**:

b Elige el factor de bloqueaje. Generalmente innecesario.

v Reporta un desplegado de las actividades realizadas por el comando.

a Toma la información de la tabla de contenido del archivo nombrado. Hasta que usted necesite extraer un archivo, la cinta en la cual se encuentra el archivo deseado no necesita ser montada.

Antes de que usted respalde sus file systems con el comando **ufsdump**, usted debe de cambiar el nivel de ejecución del sistema a modo de **single-user.**

Notificar a todos los usuarios del sistema y clientes de NFS, con los comandos **wall**, **rwall** y **shutdown** (para mayor información sobre estos comandos ver recetarios de comandos remotos y para cambio de niveles de ejecución):

En el caso del comando shutdown, el ejemplo sería el siguiente:

 # shutdown -g60 -i1 -y *"El sistema será bajado para hacer respaldos diarios"*

En donde **-g60** indica que el tiempo en el que será dado de baja el servidor será en 60 segundos, **-i1** indica que el nivel de ejecución será de single-user y la **-y** sirve para mandar un mensaje a todos los usuarios que se encuentren en sesión.

Usted puede desear checar sus file systems con el comando **fsck** antes de realizar el respaldo. Esto no es esencial, pero le puede dar a usted alguna seguridad de que su respaldo será realizado sobre una partición limpia.

Precaución - Es muy importante que sus respaldos sean realizados sobre file systems desocupados. **ufsdump** es un comando de dos pasadas. La primera pasada toma la información de los inodos, la segunda pasada toma los bloques de datos. Si un filesystem está activo mientras el **ufsdump** está corriendo, un archivo o directorio puede cambiar entre la primera y segunda pasada del **ufsdump**. El resultado final es entonces un respaldo defectuoso.

Si se deseara respaldar el file system de home en una unidad de cinta de 1/2 pulgada Fujitsu/Xylogics de alta densidad, se podrían utilizar las siguientes instrucciones:

En primer lugar deberá ser desmontado el file system de */home*, con el comando umount, de la siguiente manera:

 # umount */home*

luego la instrucción que ser mostrará a continuación permitirá realizar el resplado del file system:

 # ufsdump *0udf 6250 /dev/rmt/0 /home*

Donde **0** es un nivel de respaldo comprendido entre 0 y 9, **u** es una llave que indica que será actualizada la base de datos de **/etc/dumpdates**, **d** es una llave que indica la densidad de grabación, que en este caso será de **6250 bpi** (bits per inch -bits por pulgada-), para mayor información al respecto referirse al recetario de unidades

de cinta, **f** indica que el dispositivo que para el ejemplo es **/dev/rmt/0** y por último **/home** es el nombre de la partición a ser respaldada.

Para el mismo ejemplo, sólo que en un cartucho de 1/4 de pulgada de 600 pies, la instrucción sería la siguiente:

 # ufsump *0ucsf 600 /dev/rst5 /home*

Donde **0** es un nivel de respaldo comprendido entre 0 y 9, **u** es una llave que indica que será actualizada la base de datos de **/etc/dumpdates**, **c** indica que el respaldo será realizado hacia un cartucho de cinta, **s** indica la longitud de la cinta que en este caso es de 600 pies, **f** indica que el dispositivo que para el ejemplo es **/dev/rst5** y por último **/home** es el nombre de la partición a ser respaldada.

En el caso de que se deseara utilizar cintas de cartucho de Exabyte de 8mm de 5.0 Gbytes:

 # ufsdump *0usf 13000 /dev/rst5 /home*

Donde **0** es un nivel de respaldo comprendido entre 0 y 9, **u** es una llave que indica que será actualizada la base de datos de **/etc/dumpdates**, **s** indica la longitud de la cinta que en este caso es de 13000 pies, **f** indica que el dispositivo que para el ejemplo es **/dev/rst5** y por último **/home** es el nombre de la partición a ser respaldada.

Si se deseara relizar el respaldo usando un host remoto de cinta, la instrucción sería:

 # ufsdump *0ucf earth:/dev/rst8 /homet*

Donde **0** es un nivel de respaldo comprendido entre 0 y 9, **u** es una llave que indica que será actualizada la base de datos de **/etc/dumpdates**, **c** indica que el respaldo será realizado hacia un cartucho de cinta, **f** indica que el dispositivo que para el ejemplo es **/dev/rst8** -sólo que en este caso está precedido por el nombre **earth**

seguida de dos puntos (:) para indicar cual será la máquina de la que accesaremos su unidad de cinta- en este caso se deberá tener el archivo **/.rhosts** en el host *earth* (para mayor información ver recetario de comandos remotos) y por último **/home** es el nombre de la partición a ser respaldada.

Si se deseara realizar el respaldo de varios file systems en una misma unidades de cinta se utilizan las siguientes instrucciones:
Primero se deberá de realizar un ufsdump sin que se rebobine la cinta:

 # ufsdump *5ucsf 13000 /dev/nrst5 /export*

Donde **5** es un nivel de respaldo comprendido entre 0 y 9, **u** es una llave que indica que será actualizada la base de datos de **/etc/dumpdates**, **c** indica que el respaldo será realizado hacia un cartucho de cinta, **s** indica la longitud de la cinta que en este caso es de 13000 pies, **f** indica que el dispositivo que para el ejemplo es */dev/nrst5* -sólo que en este caso la letra n del dispositivo indica que la cinta no será rebobinada una vez que se haya realizado el respaldo- y por último **/export** es el nombre de la partición a ser respaldada.

Realizar el resplado de la partición **/home**:

 # ufsdump *5ucsf 13000 /dev/nrst5 /home*

Donde **5** es un nivel de respaldo comprendido entre 0 y 9, **u** es una llave que indica que será actualizada la base de datos de **/etc/dumpdates**, **c** indica que el respaldo será realizado hacia un cartucho de cinta, **s** indica la longitud de la cinta que en este caso es de 13000 pies, **f** indica que el dispositivo que para el ejemplo es */dev/nrst5* -sólo que en este caso la letra n del dispositivo indica que la cinta no será rebobinada una vez que se haya realizado el respaldo- y por último **/home** es el nombre de la partición a ser respaldada.

Realizar el último respaldo del file system de **/opt**:

 # ufsdump *5ucsf 13000 /dev/rst5 /opt*

Donde **5** es un nivel de respaldo comprendido entre 0 y 9, **u** es una llave que indica que será actualizada la base de datos de **/etc/dumpdates**, **c** indica que el respaldo será realizado hacia un cartucho de cinta, **s** indica la longitud de la cinta que en este caso es de 13000 pies, **f** indica que el dispositivo que para el ejemplo es ***/dev/rst5*** -sólo que en este caso sin la letra n del dispositivo, para indicar que la cinta será rebobinada una vez que se haya realizado el respaldo- y por último **/opt** es el nombre de la partición a ser respaldada.

Cuando usted está respaldando pequeños file systems, o realizando respaldos de tipo incremental y conoce que la cantidad de datos es pequeña, usted puede desear utilizar una cinta para múltiples file systems. Para que sea exitoso, usted debe de asegurarse de que haya suficiente espacio para almacenar todos los file systems deseados.

Una vez que se ha realizado el respaldo de los file systems deseados, el siguiente paso será conocer el contenido de una cinta, para lo cual se utilizará el comando ufsrestore de la siguiente forma:

Desplegar el contenido de un volumen de ufsdump

 # ufsrestore tvf */dev/rst5*

Donde **t** es una llave que indica que la cinta sólo será leída, **v** es una llave para indicar que la salida será un listado similar al del comando **ls** y **f** indica que el dispositivo que para el ejemplo es */dev/rst5*.

Para desplegar el contenido del respaldo de varios file system en una misma cinta, se utilizarán las siguientes instrucciones (en este caso se listarán los respaldos del ejemplo de respaldo múltiple que se mostró con anterioridad)
Con esta instrucción se puede ver el contenido del primer file system respaldado:

 # ufsrestore tvf */dev/nrst5*

Esta instrucción es para ver el contenido del segundo file system respaldado:

 # ufsrestore tv*f /dev/rst5*

La última instrucción es para ver el contenido del tercer file system respaldado:
 # ufsrestore tv*f /dev/rst5*

En este último ejemplo, la cinta se rebobinará al terminar de realizarse la lectura de la cinta (si no se desea que ocurra esto, usar el dispositivo **/dev/nrst5**).

Una vez que se sabe cómo visualizar el contenido de una cinta, el siguiente paso es el de extraer archivos de la misma:

Por ejemplo, si deseáramos extraer únicamente el archivo /etc/passwd del file system de root (/), el procedimiento sería el siguiente:

Cambiarse a un directorio temporal, o bien al file system correspondiente:

 # cd */tmp*

utilizar el comando ufsrestore como sigue:

 # ufsrestore *xf /dev/rst5 ./etc/passwd*

Donde **x** indica que será extraída información y **f** indica que el dispositivo de donde se obtendrá es el **/dev/rst5**, por último **./etc/passwd** es el nombre del archivo a extraerse.
En el mismo caso, si se deseara extraer archivos a través de la red, el comando sería:

 # ufsrestore *xf earth:/dev/rst5 ./etc/passwd*

la explicación de la línea es igual que la del ejemplo anetrior, sólo que en este caso la palabra **earth** seguida de dos puntos (:) antes del

nombre del dispositivo, se refiere a un host que nos compartirá su unidad de cinta.

Para restaurar sus archivos de una cinta remota, debe de existir una entrada en el archivo **/.rhosts** dentro del sistema remoto. Esto precisamente porque el comando ufsrestore es usualmente ejecutado como root, especialmente para restaurar un file system completo.

Para restaurar un filesystem completo

Desmontar la vieja partición y respaldar su contenido:

 # umount */home*
Utilizar el comando ufsdump para realizar el respaldo, como se explicón en ejemplo previos:

 # ufsdump *0csf 13000 /dev/rst4 /home*

Preparar la nueva partición corriendo el comando newfs (para mayor información referirse al tema 8), de esta manera está disponible con un montaje temporal para restaurar el filesystem:

 # newfs */dev/rdsk/c0t3d0s7*

Montar de nueva cuenta el file system:

 # mount */dev/dsk/c0t3d0s7 /home*

Cambiarse al directorio de **/home**:

 # cd */home*

Usar el comando ufsrestore de la siguiente forma:

 # ufsrestore *rvf /dev/rst4*

Donde **r** indica que será extraído todo el contenido de la cinta, **v** indica que será realizado un reporte de actividades y **f** indica que el dispositivo de donde se obtendrá es el **/dev/rst4**.

Una vez restaurado el filesystem, desmontarlo y checarlo de nueva cuenta con el comando fsck:

 # cd */*
 # umount */mnt*
 # fsck */dev/rdsk/c0t3d0s7*

Nota: Si se desea generar un nuevo file system a partir de un respaldo realizado previamente, definir el filesystem en el archivo **/etc/vfstab** y montarlo (ver tema 8).

tar

Antecedentes

El comando **tar** (tape archive) le permite hacer respaldos de todos los archivos contenidos en un directorio. Al contrario que **ufsump, tar** no está diseñado para el respaldo de file systems. Si usted utiliza **tar** para respaldar el directorio root (/), esto puede ser interpretado como el respaldo de todo el conjuntos de archivos y directorios que constituyen al sistema, incluyendo aquellos recursos que fueran montados vía **NFS. tar** no soporta respaldos de volúmen múltiple.

Procedimiento

tar c|r|t|x *[modificadores] archivo...*

llaves de **tar**:

c Esta opción permite hacer el respaldo de la información a un medio magnético.

r Escribe los archivos nombrados en el fin del archivo tar. (No soportado en cintas de 1/4 de pulgada).

x Sirve para indicar que se va a restaurar la información al disco duro.

t Indica que sólo se va a visualizar el contenido del dispositivo magnético.

modificadores del **tar**:

v Con esta opción se ven en pantalla los archivos.

f Es la opción para indicar el dispositivo sobre el cual se trabajará

b Con esta opción se indica el tamaño del bloque a manejar en el medio magnético.

B Realiza múltiples lecturas para determinar exactamente la cantidad de bytes que son necesarios para llenar un bloque. Es necesaria en el caso de que el comando **tar** se utilice a través de la red.

p Restaura los archivos con los permisos que se encuentren en la cinta.

Si se desean respaldar todos los archivos debajo del directorio actual, se utilizará la siguiente instrucción:

#tar *cvf /dev/rmt/0 **

Donde **c** indica que la información será respaldada en algún dispositivo, **v** indica que se hará un despliegue de actividades, la **f** al igual que en el comando ufsrestore sirve para indicar el dispositivo que en este caso es **/dev/rmt/0**, y por último el asterisco (*) sirve para indicar los archivos que se van a respaldar.

#tar *cvf /dev/fd0a /etc*

En este ejemplo respalda la información del directorio **/etc** a una unidad de floppy.

Para listar el contenido de un respaldo efectuado con **tar**, deberá usarse la siguiente instrucción:

#tar *tvfb /dev/rmt/0 20*

Este ejemplo lista el contenido de cinta con un factor de bloque de 20. Donde **t** indica que se realizará un lectura, **v** indica que se realizará un reportes de actividades, **f** es para especificar el dispositivo (en este

caso **/dev/rmt/0**) y **b** para indicar el factor de bloqueaje en este caso de 20 bloques.

Si se deseara extraer el contenido de una cinta se puede utilizar la siguiente instrucción con el comando tar:

#**tar** *xvf /dev/rmt/0 /etc/hosts /etc/ethers /etc/vfstab*

En donde **x** indica que se extraerá información de la cinta, **v** indica que se realizará un reportes de actividades, **f** es para especificar el dispositivo (en este caso **/dev/rmt/0**) y /etc/hosts, /etc/ethers y /etc/vfstab son los archivos a restaurar.

Si desea copiar el contenido de un directorio del disco duro a otro, puede utilizar la siguiente instrucción:

cd */etc* ; **tar** *cf - .* | *(***cd** */prueba*; **tar** *xfBp -)*

donde **cd** */etc* indica un cambio al directorio del cual será extraída la información, **tar** con las opciones y argumentos "*cf - .*" indica que la información será enviada a la salida estándar del sistema (para mayor información sobre el concepto de salida estándar, entrada estándar y error estándar, consultar un manual de UNIX básico); la siguiente parte de la instrucción indica un cambio al directorio */prueba*, y el comando **tar** con las opciones y argumentos "*xfBp -* " indican que la información será extraída de la entrada estándar (*xf*), se realizarán las lecturas necesarias para llenar exactamente los bloques (*B*) y se preservarán los permisos originales de los archivos (*p*).

Si se desea grabar archivos sin una trayectoria absoluta, y éstos se encuentran en diferentes directorios, se puede utilizar el siguiente ejemplo:

tar *c -C /usr include -C /etc*

la opción *c* indica que será grabada alguna información, y como no se indica el dispositivo en el cual será almacenada la información el sistema asume el valor que tenga configurado como default, *-C* indica.

cambios a los directorios */usr* y */etc*, *include* y "." indican los archivos a ser respaldados.

cpio

Antecedentes

El comando cpio es un comando de respaldo estándar en UNIX, y es muy utilizado para el respaldo de archivos de que se encuentran dispersos dentro de los directorios de los usuarios.

En el caso de **cpio**, sucede algo similar que con tar, pues se trata del mismo comando tanto para respaldar, así como para restaurar información.

La diferencia con relación al tar es que se utiliza de una manera diferente, ya que cpio necesita de otro comando para efectuar el respaldo.

Procedimiento

cpio [opciones] *direccionamiento archivo(s)*

En el caso de tratarse de un respaldo, requiere de cualquiera de los siguientes comandos:

ls	(Nota: para mayor información acerca de éstos comandos, referirse a un manual de UNIX básico)
find	

Opciones:

-i Sirve para restaurar información.

-o Sirve para respaldar información

-t Al igual que con el tar sirve para ver los archivos que contiene el dispositivo.

-d	Si es necesario cpio generará los directorios no existentes
-B	Asigna un factor de bloque de 5120 bytes.
-v	Sirve para ver los archivos en la pantalla, en conjunto con las opciones anteriores (**-i, -o**).
-p	Toma los archivos nombrados y copia o liga (opción -l) hacia el nuevo directorio.
-l	Liga los archivos al directorio indiado (sólo con la opción -p)
-m	Retiene la fecha de modificación previa de los archivos, no funciona con directorios.

% **ls** */usr2/juan/** | **cpio -ov** >*/dev/rmt/0*

En este caso se hace un respaldo del contenido del directorio juan, enviándose a una unidad de cinta. La opción **-o** sirve para indicarle al comando **cpio** que se grabará la información en algún dispositivo, en este caso en el **/dev/rmt/0**, la opción **-v** sirve para que sea reportado en pantalla un despligue de las actividades ejecutadas por el comando.

Si se deseara leer el contenido de un diskette, se usará la siguiente instrucción:

%cpio *-itv* < */dev/fd0a*

aquí se visualizará únicamente el contenido de la unidad de floppy. La opción **-i** indica que el comando **cpio** recibirá la entrada de algún dispositivo, la **-v** funciona igual que en el ejemplo anterior y la **-t** le indica al comando que la salida únicamente será en pantalla -si fuera omitida esta opción la información sería restaurada en el disco duro.

Si en realidad se desea restaurar la información en el disco duro de la máquina, deberá teclearse la siguiente instrucción:

% **cpio** *-iv < /dev/fd0a*

En este ejemplo se restaura la información del floppy, a través de las opción **-i** indica que el comando **cpio** recibirá la entrada de algún dispositivo, la **-v** funciona igual que en el ejemplo anterior

Si se desea respaldar todos los archivos que comiencen con "a" del usuario odilia, se hará:

% **ls** */usr2/Morgan/odilia/a** | **cpio***-ov >/dev/rst4*

/usr2/Morgan/judith/aaa
/usr2/Morgan/judith/arch2.1.
/usr2/Morgan/judith/arch3.1
/usr2/Morgan/judith/arch4.1
usr2/Morgan/judith/arch5.1
/usr2/Morgan/judith/archivoA
/usr2/Morgan/judith/audio-file.au
/usr2/Morgan/judith/audio.au
35 blocks

Si se desea hacer un archivo de respaldo con el comando cpio se puede utilizar la siguiente instrucción:

ls | **cpio** *-oc > ../respaldo1*

donde **ls** tiene por objeto listar el contenido del directorio actual y mandar dicha información al comando **cpio**, quien a su vez con las opciones *oc* tomará la acción de grabar dichos archivos en el archivo de nombre *respaldo1*.

Si se desea respaldar un directorio con todos sus posibles subdirectorios y archivos y enviarlos a uno nuevo, se puede utilizar la siguiente línea de comando:

 # find . *-depth -print* | **cpio** *-pdlmv /dirnuevo*

donde el comando **find** tiene por objeto listar todos los archivos y directorios debajo del actual y **cpio** recibirá esa información con el objeto de identificar la información a copiar, la opción *p* sirve para indicar que los archivos serán enviados hacia un nuevo directorio (*/dirnuevo*), la opción *d* indica que en caso de ser necesario el comando **cpio** generá los subdirectorios que sean necesarios, la opción *l* indica que los archivos en caso de ser posible serán ligados al nuevo directorio, la opción *m* sirve para preservar la fecha de última modificación y la opción *v* sirve para ver en pantalla un reportes de las operaciones que van siendo ejecutadas.

dd.

Antecedentes

El comando **dd** es usado para convertir y copiar archivos con varios formatos de datos. Y una especial capacidad para leer o escribir en dispositivos de cinta. Otra de sus principales características es que tiene la posibilidad de extraer datos crudos (byte por byte) de un dispositivo determinado, con lo cual se garantiza que la información podrá ser accesada independientemente del formato en el cual venga respaldada.

El comando **dd** es muy poderoso, ya que a través de este se pueden realizar respaldos remotos, en el caso de que en la máquina local no se posean dispositivos de almacenamiento.

Procedimiento

dd *[opción=valor]* ...

opciones del **dd**:

if=nombre El archivo de entrada (**if**) es tomado del **nombre**, el cual puede ser un archivo o un dispositivo. La entrada estándar es el default.

of=nombre El archivo de salida (**of**) es tomando del **nombre**. La salida estándar es el default.

bs=n El factor de bloqueaje tanto de entrada (**bs**) como de salida (**obs**) es **n**. El valor por default es 512 bytes. Para indicar que sean kilobytes se puede utilizar la letra **k** al número (por ejemplo, bs=24k), o bloques de 512 bytes con una **b** (por ejemplo, bs=204b)

Si se desea realizar un respaldo de una partción de un disco duro a otra, puede ser utilizada la siguiente instrucción:

 # dd *if=/dev/dsk/c0t3d0s5 of=/dev/dsk/c0t1d0s4*

donde como argumentos del comando **dd,** están **if** que tiene por función indicar de qué dispositivo sera accesada la información (en este caso de la partición 5 del disco 3, para mayor información consultar el recetario de instalación física del equipo Sun) y por otra parte **of** que a su vez tiene la función de indicar el dispositivo de salida para el almacenamiento de la información -en este caso será la partición 4 del disco duro 1-.

Asímismo el respaldo puede ser realizado entre una partición de un disco duro y una unidad de cinta, por ejemplo:

 # dd *if=/dev/dsk/c0t3d0s5 of=/dev/rst4*

donde como argumentos del comando **dd,** están **if** que tiene por función indicar de qué dispositivo sera accesada la información (en este caso de la partición 5 del disco 3, para mayor información consultar el recetario de instalación física del equipo Sun) y por otra parte **of** que a su vez tiene la función de indicar el dispositivo de salida para el almacenamiento de la información -en este caso será la unidad de cinta con la dirección lógica número 4-.

Como se mencionó anteriormente se pueden realizar respaldos de forma remota auxiliándose con el comando dd, y el siguiente procedimiento es un claro ejemplo de ello:

 # tar *cvf - /etc* **| rsh** *unam* **dd** *of=/dev/rmt/0*

en este caso se supone que en el servidor local no se posee una unidad de cinta o bien no puede ser accesada en ese momento, y es por ello que se requiere realizar el respaldo en la unidad de cinta del servidor **unam**. La explicación de la instrucción sería la siguiente, se efectúa el comando de respaldo tar en el servidor local para indicar el formato del mismo, sólo que en este caso en lugar de mencionar

algún dispositivo se pone un guión "-", la salida del comando **tar** será redireccionada a través de un "|" y servirá como entrada para el comando rsh que a su vez ejecutará el **dd** en el servidor remoto **unam** tomando como entrada del comando la estándar y como salida el dispositivo **/dev/rst4**.

Si se deseara restaurar el respaldo remoto, deberá ejecutarse la siguiente instrucción:

 # rsh *unam* **dd** *if=/dev/rmt/0* | **tar** *xvfB* -

en donde el comando **dd** sirve para accesar el dispositivo remoto en el servidor **unam**, denotado por **if=/dev/rmt/0**, en este caso extrae la información de la unidad de cinta y por otra parte el comando **tar** recibe la información generada por el **dd** y la baja a disco, ya que las opciones **xvfB** en base a su orden de aparición tienen por función extraer la información del respaldo, despliegue de actividades, indicación del dispositivo y por último se le indica al comando que tendrá que realizar varias lecturas en el caso de que se detecte algún tipo de inconsistencia en la estructura de la información -con el objeto de leer la cantidad completa de bytes para llenar un bloque-.

Capacidades de almacenamiento en cinta.

Cinta de 1/2 pulgada, 2400 pies, 9-tracks para dispositivo de cinta con interface Xylogics:

dispositivo	densidad	capacidad
rmt0	1600bpi	30MB
rmt8	6250bpi	150MB

Cinta de 1/2 pulgada, 2400 pies, 9-tracks para dispositivo de cinta con interface SCSI:

dispositivo	densidad	capacidad
rst5	800bpi	15MB
rst8	1600bpi	30MB
rst16	6250bpi	150MB

Cinta de 1/4 pulgada, 1000-bpi para cintas de cartucho:

dispositivo	format	tracks	longitud	capacidad
rst5	QIC-11	4	450 pies	20MB
rst8	QIC-24	9	450 pies	45MB
rst8	QIC-24	9	600 pies	60MB
rst5	QIC-150	18	600 pies	150MB

Cartuchos de cinta de tipo **Exabyte** con capacidad de 2.3GB, utilizan el dispositivo **rst5**.

Este comando es utilizado tanto para respaldar, así como para restaurar información, lo que varía para cada acción son las opciones utilizadas.

Apéndice K. Adicionar área de swap

Apéndice K. Adicionar área de swap

Antecedentes:

Otro de los problemas comunes dentro de los sistemas de cómputo en red es la falta de memoria en *RAM* para la ejecución de todos los procesos que por parte de los usuarios son requeridos; es aqui donde los nuevos sistemas operativos brindan nuevas alternativas a esta limitante a través de la implantación del concepto de la memoria virtual -que en pocas palabras se constituye por el total de la memoria que se tiene en *RAM* más el espacio es disco destinado a un área que se denomina *swap*-. En algunos casos la memoria física de los equipos no es suficiente, y no se cuenta con los recursos económicos como para adquirir memoria, en otros aunado al problema anterior, se cuenta con que el área que se destino para swap en el disco al momento de hacer la instalación del sistema operativo es insuficiente.

Comandos a utilizar:

Comandos básicos	Comandos de administración
df	**swap**
cd	**mkfile**
ls	

Procedimiento:

Una vez que se conocen los comandos y archivos a utilizar, es cuando se puede adicionar espacio en swap a través del siguiente procedimiento:

1. Para determinar cuanto es el uso de *swap* en el sistema se pueden utilizar los siguientes comandos:

```
# df -k
Filesystem            kbytes    used   avail capacity  Mounted on
/dev/dsk/c0t3d0s0      43423   26805   12278    69%    /
/dev/dsk/c0t3d0s6     226759  197134    6955    97%    /usr
/proc                      0       0       0     0%    /proc
fd                         0       0       0     0%    /dev/fd
/dev/dsk/c0t3d0s3      19183    7561    9712    44%    /var
swap                  122876  122652     224    99%    /tmp
/dev/dsk/c0t3d0s5     143641   94924   34357    73%    /opt
/dev/dsk/c0t3d0s7     480815  151593  281142    35%    /pypsa
/dev/dsk/c0t0d0s4     288391  122754  136807    47%    /usr/acad
/dev/dsk/c0t0d0s5     288391  187153   72408    72%    /iiasa
/dev/dsk/c0t0d0s6     101007    3084   87823     3%    /sparc20
/dev/dsk/c0t1d0s6     963662  595967  271335    69%    /iiasa2
/dev/dsk/c0t4d0s2     537784  365387  118627    75%    /usr2
#
```

que como hemos visto con anterioridad nos muestra el porcentaje de saturación de los diferentes file systems de la estación de trabajo. O bien con

```
# swap -l
swapfile             dev    swaplo   blocks   free
/dev/dsk/c0t3d0s1    32,25      8   102808     98
/dev/dsk/c0t0d0s1    32,1       8   102808     64
#
```

que es un comando específicamente creado para el monitoreo del área de *swap* y la opción -l indica que la función del comando será listar o desplegar estadísticas del area de *swap*. En ambos casos (con el comando **df** y con el comando **swap**) las estadísticas denotan que el sistema está haciendo gran uso del área de *swap*, y que un momento determinado dicha área puede ser insuficiente para albergar las imágenes de todos los procesos del sistema, lo que podría ocasionar una caída del mismo.

2. Para adicionar un área adicional de *swap* a las ya existentes, primero deberá de crearse un archivo para dicho fin, en ese caso, checar ¿en qué file system puede ser generado uno del tamaño que se desea?:

 # df *-k*

Filesystem	*kbytes*	*used*	*avail*	*capacity*	*Mounted*
/dev/dsk/c0t3d0s0	*43423*	*26805*	*12278*	*69%*	*/*
/dev/dsk/c0t3d0s6	*226759*	*197134*	*6955*	*97%*	*/usr*
/proc	*0*	*0*	*0*	*0%*	*/proc*
fd	*0*	*0*	*0*	*0%*	*/dev/fd*
/dev/dsk/c0t3d0s3	*19183*	*7561*	*9712*	*44%*	*/var*
swap	*122876*	*122652*	*224*	*99%*	*/tmp*
/dev/dsk/c0t3d0s5	*143641*	*94924*	*34357*	*73%*	*/opt*
/dev/dsk/c0t3d0s7	*480815*	*151593*	*281142*	*35%*	*/pypsa*
/dev/dsk/c0t0d0s4	*288391*	*122754*	*136807*	*47%*	*/usr/acad*
/dev/dsk/c0t0d0s5	*288391*	*187153*	*72408*	*72%*	*/iiasa*
/dev/dsk/c0t0d0s6	*101007*	*3084*	*87823*	*3%*	*/sparc20*
/dev/dsk/c0t1d0s6	*963662*	*595967*	*271335*	*69%*	*/iiasa2*
/dev/dsk/c0t4d0s2	*537784*	*365387*	*118627*	*75%*	*/usr2*

#

aquí por ejemplo, se puede utilizar el file system de */sparc20* para tal motivo, ya que tiene alrededor de 87 Mb disponibles

3. Una vez determinado el file system, pasarse al directorio de montaje y crear el archivo:

 # cd */sparc20*

con el comando **cd** cambió al directorio */sparc20*

 # mkfile *40m swap3*

con el comando **mkfile** se generó un archivo que pueda ser utilizado como un área de *swap* de 40 Mb llamado *swap3*

ls *-l swap3*
-rw------T 1 root other 41943040 Nov 8 16:34 swap3
#

para verificar que el tamaño del archivo que deseamos fué correcto, se puede utilizar el comando **ls** con la opción **-l** con el objeto de que también reporte el tamaño del archivo en bytes.

4. Con los pasos anteriores tenemos tan sólo reservada una nueva área para *swap*, pero ahora lo que se debe de hacer es adicionarla a las ya existentes con el mismo comando **swap**.

 # swap *-a /sparc20/swap3*

la opción **-a** del comando **swap** indica que será añadido el archivo *swap3* al espacio en *swap*

```
# swap -l
swapfile             dev    swaplo  blocks   free
/dev/dsk/c0t3d0s1    32,25  8       102808   36808
/dev/dsk/c0t0d0s1    32,1   8       102808   35544
/sparc20/swap3       -      8       81912    31912
#
```

la opción **-l** del comando **swap** sirve, como se indicó anteriormente, para adicionar listar o desplegar estadísticas del área de *swap*.

Apéndice M
Instalar *terminales X*

Apéndice M. Instalar *terminales X*

Antecedentes:

En una *red*, siempre se está expuesto, a instalar nuevo equipo, o adicionar máquinas del mismo tipo, si esto sucede no hay mayor complicación, pues todos los servicios están presentes. En caso contrario, se tienen que activar los servicios, en está receta suponemos que no existen los servicios para una *terminal X*, modelo *15b de NCD*.

Comandos a utilizar:

Comandos básicos

- **mkdir**
- **cd**
- **cp**
- **vi**
- **touch**
- **chmod**
- **ps**

Comando de administración

- **tar**
- **mount**

Procedimiento:

1.- Realizar la instalación física de las *terminales X*

Verifique: El modelo de la terminal; *transceiver*, *segmento de red*; cables de corriente; *concentrador*, y *servidor* a donde se conectará la terminal, y donde se activarán los servicios.

2.- Determinar el tipo de media (*cinta, cdrom*, etc), del cuál será accesado el *software* para las *terminales X.* en nuestro caso utilizaremos el *cdrom.*

3.- Introducir el disco compacto en el dispositivo de *cdrom*

Si no se encuentra *montado* o activado
3.1 crear un directorio llamado *cdrom*

mkdir *cdrom*

Realizar el montaje del *cdrom* al directorio */cdrom*, a través del siguiente instrucción:

mount *-rt hsfs /dev/sr0 /cdrom*

o bien:

mount *-F hsfs -0 r0 /dev/dsk/c0/t6/d0/s0 /cdrom*

4.- Verefique el espacio en disco, utilice el comando **df.**

5.- Crear un directorio temporal para nuestra instalación, siempre se recomienda que dicho directorio se haga a la altura de ***/home/servidor*** y se denomine *NCD*, quedando así :

mkdir */home/frida/NCD*

6.- Verificar que exista un directorio de *montaje* para el *cdrom*, por default en algunos sistemas lo podemos localizar a la altura de la *raíz* del sistema denominado como *cdrom.*

Si no existe debemos crearlo (vea el paso 3, de está receta). En el caso de que estemos trabajando con una cinta, este directorio de montaje no es necesario y bastará sólo con el directorio de trabajo del que hablamos en el punto anterior.

7.- Copiar el archivo 0 del *cdrom*, al directorio temporal.

cp */cdrom/ncdware_3_0_1standard/generic/0 /home/frida/NCD*

8.- Hagamos un cambio de directorio.

cd */home/frida/NCD*

9.-. Aplicar el comando **tar** al archivo 0 (cero).

tar *xvf* 0

10.- Correr el archivo denominado ***ncdinstall***

./ncdinstall

El cuál es el *scrip* de instalación del software necesario para levantar los servicios de la *terminales X*. Para mayor detalle al respecto vea tema **Servicios requeridos en la instalación de una terminal X.**

11.- Una vez que el *scrip* de instalación haya terminado de bajar el *software*, el administrador de la *red* puede iniciar la configuración de las terminales.

Configuración de una *terminal X*

1.- Modifique el archivo ***inetd.conf***, que se encuentra bajo el directorio ***etc*** , primero haga un cambio de directorio, y utilice el editor **vi**, teclee las iguientes instrucciones:

cd */etc*
vi *inetd.conf*

2.- Busque la línea siguiente:

 # tftp dgram udp wait root /usr/etc/in.tftpd in.tftpd -s /tftpboot

3.- Descomente la línea, quitando (el símbolo de gato) # y además eliminar la parte que activa el seguro, llamado *secure tftp*

Elimine

-s /tftpboot

La línea debe quedar así:

tftp dgram udp wait root /usr/etc/in.tftpd in.tftpd

4.- Dar de alta las *direcciones Ethernet* e *Internet*, de la terminal que será instalada, en los archivos *hosts* y *ethers*, respectivamente, no se olvide que están bajo el directorio ***/etc***.

Veamos el archivo *hosts*, utilizando el editor **vi**.

vi */hosts*

Insertemos la siguiente línea:

192.9.200.12 javier

Guarde los cambios y salga del archivo. Asignamos a la terminal su *dirección lógica* que es *192.9.200.100* y un nombre que es javier, para identificar rápido a la terminal.

Ahora veámos el archivo *ethers*, para agregar la *dirección física* de la terminal llamada javier.

vi *ethers*

Insertemos la siguiente línea:

00:00:a7:11:cf:78 javier

Guarde los cambios y salga del archivo. Asignamos a la terminal su *dirección física que es 00:00:a7:11:cf:78* y un nombre que es javier, para identificar rápido a la terminal.

5.- En base a la *dirección Internet* asignada a cada una de las terminales,. deberá crearse un archivo en el directorio **/usr/lib/X11/ncd/configs** de la siguiente forma:

Determine primero el nombre del archivo a partir de su *dirección lógica* o *Internet*, para la terminal javier está es *192.9.200.12*

Tomar el primer grupo de dígitos y convertirlos a hexadecimal

Decimal	Hexadecimal
192	C0
9	09
200	C8
12	0C

El nombre del archivo, debe estar en mayúsculas, quedando de la siguiente forma:

C009C80C

Para crear el archivo, utilizaremos el comando **touch**, recuerde que deberá crearse en el directorio *configs* con la siguiente instrucción:

touch */usr/lib/X11/ncd/configs/C009C80C*

6.- Cambiar los permisos de acceso al archivo con la siguiente instrucción:

chmod *666 C009C80C*

Ahora el archivo tiene permisos de lectura y escritura para todos, los usuarios del sistema.

7.- Reinicializar el *servidor* y una vez que haya levantado todos los servicios verificar que estén corriendo el proceso ***in.tftpd***, con ayuda del comando **ps**

ps *-eda | rarpd*

Apéndice N
Comunicar *servidores*, utilizando mail.

Apéndice N. Comunicar *servidores*, utilizando mail.

Antecedentes:

Comando: **mail**, es un programa interactivo y flexible para componer, enviar y recibir correo o mensajes entre usuarios. La forma de activar **mail** es :

frida% **mail** *nombre_de_usuario*

Siempre que el nombre del usuario, a quien va dirigido y el remitente, existan en el *servidor* local, en esta caso ambos usuarios son locales a frida.

Comandos a utilizar:

Comandos básicos

mail
cd
cat
vi
cp

Mail, como tal tiene varias opciones algunas de ellas son:

mail *-h* Presenta las cabeceras de los mensajes, empezando por el más reciente al mas antiguo. Quedando dentro de **mail**.

mail *-p* Presenta todos los mensajes que hay para un usuario, y regresa al *prompt*, del *sistema operativo*.

Veamos, como configurar un *servidor* de **mail**, para que usuarios locales de un *servidor*, puedan comunicarse, con otros usuarios de un *servidor* diferente. Es importante recordar que un *servidor* local, es aquella máquina donde yo usuario, tengo acceso al sistema, ahora bien cuando hablamos de un *servidor* remoto, es una máquina que existe en la *red*, y que también puedo accesar bajo, ciertas restricciones.

Para comunicación entre usuarios locales **mail**, es ideal, y no se necesita de configurar archivos, a menos que la comunicación sea entre usuarios remotos que están dados de alta en diferentes *servidores*, para lo cual es necesario configurar un *servidor* de **mail**.

Procedimiento:

1.- Determine cual de sus *servidores*, será el *servidor* de **mail**.

En este caso frida será el *servidor* de **mail**.

2.- En cada servidor (excepto en frida), vaya al archivo de *hosts*, que se localiza bajo el directorio */etc*.

cd */etc*

Despleguemos el contenido de dicho archivo:

cat *etc*

Vera algo así:

192.9.200.200	***frida***
192.9.200.201	***diego***
192.9.200.202	***sandy***
192.9.200.203	***alix***

3.- Ahora utilizando el editor **vi**, de *UNIX*,

vi *hosts*

Teclee delante del nombre de frida , *mailhost*, debe quedar algo así:

192.9.200.200	***frida***	***mailhost***
192.9.200.201	***diego***	
192.9.200.202	***sandy***	
192.9.200.203	***alix***	

4. Con esto se termina de configurar a los no *servidores* de **mail**, y pase al punto 7 . Si su *servidor* es el *servidor* de **mail** pase al siguiente punto.

5.- Cambio de directorio.

cd */etc/mail*

6.- Hay que copiar el archivo *main.cf*, que activa la comunicación entre *servidores* a el archivo *sendmail.cf*

cp *main.cf sendmail.cf*

Definimos así, la comunicación entre *servidores*, por medio de **mail**.

7.- Si se quiere comunicar con un usuario, que se encuentra en servidor diferente del suyo, teclee lo siguiente:

frida% **mail** *nombre_de_usuario@nombre_servidor*

Nota: Este apéndice, no tiene por finalidad explicar el funcionamiento de **mail**, más sin en cambio si el de lograr comunicar usuarios de *servidor local* y uno o varios *remotos*.

Apéndice O
Modificar un *servidor* de la *red*.

Apéndice O. Modificar un servidor de la *red.*

Antecedentes:

En casos particulares que se llega a la unificación de varias *redes* pequeñas o locales en una sola, hablamos de una empresa x con empresa y o bien, las *redes* de diferentes deptos. por diversas razones, resulta que dos *servidores* (o *hosts*) tienen el mismo nombre, y recordemos que este nombre debe ser único, por funcionalidad, y para evitar problemas en la *red.*

Comandos a utilizar:

Comandos básicos
cd
vi

Comando de administración
ifconfig

Procedimiento:

Ante está situación la respuesta es cambiarle el nombre a uno de los dos *servidores*, el procedimiento es el siguiente:
1.- Ambos *servidores* tienen por nombre frida

2.- Debemos modificar varios archivos

/etc/nodename
/etc/inet/hosts
/etc/hostname.le0
/etc/net/ticlts/hosts
/etc/net/ticots/hosts
/etc/net/ticotsord/hosts

Recuerde que su localización de estos archivos es bajo el directorio */etc.*

3.- Utilizaremos el editor de *UNIX*, que es **vi** (*visual*)

4.- Teclee **vi** */etc/nodename*

A continuación vera el nombre de frida, que el nombre correspondiente al *servidor*, cambie este nombre por el de diego, que será el nuevo nombre para el *servidor.*

5.- Guarde los cambios y salga del archivo.

6.- Realice el paso 4 y 5 en el resto de los archivos antes mencionados en el paso 2.

7.- Si quiere modificar la dirección del *servidor* diego (antes frida), vuela a el archivo *hosts*, utilizando el editor **vi**, de *UNIX*,

vi *hosts*

Vera algo así:

192.9.200.200	***frida***
192.9.200.201	***diego***
192.9.200.202	***sandy***
192.9.200.203	***alix***

Teclee la nueva dirección para el *servidor* diego, que será *192.9.200.211*, reemplazando la anterior.

192.9.200.200	***frida***
192.9.200.211	***diego***
192.9.200.202	***sandy***
192.9.200.203	***alix***

8.- Guarde los cambios y salga del archivo.

9.- Teclee el siguiente comando, para asignar la *interface* de *red le0* a la nueva dirección:

ifconfig *le0 192.9.200.211*

Verifique que los cambios se realizarón con el siguiente comando.

ifconfig *-a*

10.- Reinicializar el sistema para activar el nuevo nombre del *servidor.*

Apéndice P
Realizar el *montaje* del *cdrom*

Apéndice P. Realizar el *montaje* del *cdrom*

Antecedentes:

En algunos casos, necesitamos que el *servidor* reconozca un dispositivo nuevo, como sería el caso del *cdrom*, una manera sencilla, puede ser ejecutar un **reboot** - *–r* , para que el *kernel*, lo acepte. Pero existe un problema, al hacer un **reboot**, a pagamos temporalmente al *servidor*, y probablemente hay tareas o procesos que en ese momento no se pueden interrumpir, en está situación podemos realizar un *montaje*, y no detener los procesos que se ejecutan en el *servidor*.

Comandos a utilizar:

Comandos básicos	Comando de administración
mkdir	**mount**

Ante está situación la respuesta, el procedimiento es el siguiente:

Procedimiento:

1.- Verificar que exista un directorio de *montaje* para el *cdrom*, lo podemos localizar a la altura de la raíz denominado *cdrom*.

Si no existe debemos crearlo, utilice el comando **mkdir**.

mkdir */cdrom*

2.- Realizar el *montaje* del *cdrom* al directorio */cdrom*, utilizando el comando **mount**

mount *-F hsfs 0 r0 /dev/dsk/c0t6d0s0 /cdrom*

 # chmod *600 /dev/term/a*
Cambiamos el dueño a el archivo, y le damos permisos de lectura y escritura sólo para el dueño del archivo, es decir para *root.*

Si desea configurar el *puerto paralelo*, teclee lo siguiente:

 # chown *lp /dev/bpp0*
 # chmod *600 /dev/bpp0*

Cambiamos el dueño a el archivo, y le damos permisos de lectura y escritura sólo para el dueño del archivo, es decir para root.

3.- Utilizar el comando **lpadmin** para adicionar la impresora y relacionarla con un puerto de impresión.

Si se desea instalar una impresora serial, teclee lo siguiente:

 # lpadmin *-p nombre_impresora -v /dev/term/a*

Si se desea instalar una impresora paralela, teclee lo siguiente:

 # lpadmin ***-p*** *nombre_impresora* ***-v*** */dev/bpp0*

Donde:

-p Indica el nombre de la impresora

-v Indica el nombre del puerto utilizado por la impresora

Este comando **lpadmin** además registra el nombre de la impresora con el servicio de impresión.

4.- Asociar la impresora con el tipo de contenido.

 # lpadmin ***-p*** *nombre_impresora -I simple*

-I Indica el tipo de contenido

Si el tipo de contenido no está especificado, el sistema utiliza el tipo simple, lo cual significa que la impresora sólo podrá manipular contenidos *ASCII*.

5.- Elegir la línea de parámetros apropiada para la impresora

 # lpadmin ***-p*** *nombre_impresora* ***-o*** *"stty=´1200 evenp´"*

6.- Asociar la impresora con un tipo de impresora, si es necesario. Esto habilita el programa de interface para lograr una mejor inicialización antes de que se tenga alguna petición.

 # lpadmin ***-p*** *nombre_impresora* ***-T*** *proprinter*

7.- Permitir que la impresora acepte peticiones y habilite la impresión

 # accept *nombre_impresora*
 # enable *nombre_impresora*

Apéndice R
Configurar una impresora local (PostScrip)

Apéndice R. Configurar una impresora local (*PostScrip*).

Antecedentes:

Es importante mencionar , que en un sistema no siempre se tiene a disposición el ambiente gráfico, que sería lo ideal, para configurar una impresora local *PostScrip*. Por esto, se mencionará una configuración manual, considerando que se presente una situación como la antes mencionada. Recuerde que usted debe ser el súper usuario.

Comandos a utilizar:

Comandos básicos	Comando de administración
chmod	**chown** **lpadmin** **lpfilter** **accept** **enable**

Procedimiento:

1.- Verifique que usted es el súper usuario (*root*) . Asegurarse que el directorio ***/usr/lib*** se encuentra en su variable *PATH*.

 # PATH=$PATH:/usr/lib

2.- Cambiar la propiedad y el conjunto de permisos del puerto de la impresora.

Si desea configurar el *puerto serial*, teclee lo siguiente:

 # chown *lp /dev/term/a*
 # chmod *600 /dev/term/a*

Cambiamos el dueño a el archivo, y le damos permisos de lectura y escritura sólo para el dueño del archivo, es decir para *root*.

Si desea configurar el *puerto paralelo*, teclee lo siguiente:

 # chown *lp /dev/bpp0*
 # chmod *600 /dev/bpp0*

Cambiamos el dueño a el archivo, y le damos permisos de lectura y escritura sólo para el dueño del archivo, es decir para *root*.

3.- Utilizar el comando **lpadmin** para adicionar la impresora y relacionarla con un puerto de impresión.

Si se desea instalar una impresora serial, teclee lo siguiente:

 # lpadmin *-p nombre_impresora -v /dev/term/a*

Si se desea instalar una *impresora paralela*, teclee lo siguiente:

 # lpadmin ***-p*** *nombre_impresora* ***-v*** */dev/bpp0*

Donde:

-p Indica el nombre de la impresora

-v Indica el nombre del puerto utilizado por la impresora

Este comando **lpadmin** además registra el nombre de la impresora con el servicio de impresión. Desde ahora, podemos usar este nombre para indentificar a la misma impresora.

4.- Asociar la impresora con el tipo de contenido y con un tipo de impresora, si es necesario. Esto habilita el programa de interface para lograr una mejor inicialización antes de que se tenga alguna petición.

 # lpadmin -*p* *nombre_impresora* **-*I*** *PS*
 # lpadmin -*p* *nombre_impresora* **-*T*** *PS*

Donde:

-I Indica el tipo de contenido

-v Indica el tipo de impresora

5.- Usar el comando **lpfilter** para registrar los filtros *PostScript*.

 # cd */etc/lp/fd*
 # lpfilter *-f download -F download.fd*
 # lpfilter *-f dpost -F dpost.fd*
 # lpfilter *-f postdaisy -F postdaisy.fd*
 # lpfilter *-f postdmd -F postdmd.fd*
 # lpfilter *-f postio -F postio.fd*
 # lpfilter *-f postior -F postior.fd*
 # lpfilter *-f postplot -F postplot.fd*
 # lpfilter *-f postprint -F postprint.fd*
 # lpfilter *-f postreverse -F postreverse.fd*
 # lpfilter *-f posttek -F posttek.fd*

6.- Permitir que la impresora acepte peticiones y habilite la impresión

 # accept *nombre_impresora*
 # enable *nombre_impresora*

Apéndice S
Configurar una impresora remota (Servidor BSD (Berkeley Standard Distribution)).

Apéndice S. Configurar una impresora remota (Servidor BSD (Berkeley Standard Distribution)).

Antecedentes:

Cuando *clientes* de impresión de *Solaris* 2.X envían archivos a una impresora remota *PostScript*, no existe forma de especificar un contenido *PostScript* o el tipo de impresora. No se puede tener registrado ningún filtro.

Comandos a utilizar:

Comando de administración

lpsystem
lpadmin
accept
enable

Procedimiento:

1.- Asegúrese que usted es el súper usuario (*root*). Vamos a utilizar el comando **lpsystem** para registrar el nombre del *servidor* de impresión con el tipo de servicio de impresión.

 # lpsystem *-t bsd nombre_servidor*

Donde:

-t Indica versión de *UNIX* que está corriendo en el *servidor*

2.- Ahora usaremos el comando **lpadmin** para adicionar la impresora al sistema y para crear un nombre local para la misma.

 # lpadmin ***-p*** *nombre_impresora* ***-s*** *servidor_remoto*

Donde:

-p Indica el nombre de la impresora

-s Indica el nombre del *servidor* remoto

3.- Usar el comando **lpadmin** para especificar el contenido y tipo de la impresora.

 # lpadmin ***-p*** *nombre_impresora* ***-T*** *tipo_impresora* ***-I*** *tipo_contenido*

4.- Permitir que la impresora acepte peticiones y habilite la impresión

 # accept *nombre_impresora*
 # enable *nombre_impresora*

Apéndice T.
Configurar una impresora remota (Servidor SV (System V)).

Apéndice T. Configurar una impresora remota (Servidor SV (System V)).

Antecedentes:

Cuando *clientes* de impresión de *Solaris* 2.X envían archivos a una impresora remota *PostScript*, no existe forma de especificar un contenido *PostScript* o el tipo de impresora. No se puede tener registrado ningún filtro.

Comandos a utilizar:

Comando de administración

lpsystem
lpadmin
accept
enable

Procedimiento:

1.- Asegúrese que usted es el súper usuario (*root*). Vamos a utilizar el comando **lpsystem** para registrar el nombre del *servidor* de impresión con el tipo de servicio de impresión.

 # lpsystem *-t s5 nombre_servidor*

Donde:

-t Indica versión de *UNIX* que está corriendo en el *servidor*

2.- Ahora usaremos el comando **lpadmin** para adicionar la impresora al sistema y para crear un nombre local para la misma.

 # lpadmin ***-p*** *nombre_impresora* ***-s*** *servidor_remoto*

Donde:

-p Indica el nombre de la impresora

-s Indica el nombre del servidor remoto

3.- Usar el comando **lpadmin** para especificar el contenido y tipo de la impresora.

 # lpadmin ***-p*** *nombre_impresora* ***-T*** *tipo_impresora* ***-I*** *tipo_contenido*

4.- Permitir que la impresora acepte peticiones y habilite la impresión

 # accept *nombre_impresora*
 # enable *nombre_impresora*

Apendice U
Recomendaciones para conexión a la *red Internet* mediante el *nodo* del ITESM

Recomendaciones para conexión a la *red Internet* mediante el *nodo* del ITESM

Antecedentes:

El CONACYT desplegó un aviso el día 24 de Mayo de 1994 informando a la comunidad universitaria y profesional de México la integración de la **Red Tecnológica Nacional** ("Las *redes* de transmisión electrónicas, una facilidad actual sin precedentes y sin fronteras"; Excélsior p.p 27-A), informando que las universidades mas importantes del país, se interconectarán a un backbone nacional para tener un acceso a *Internet* por parte de las Instituciones Mexicanas de todo tipo.

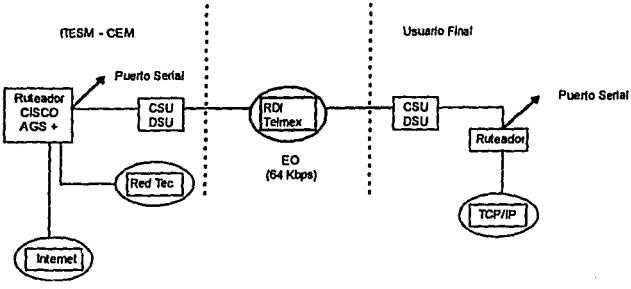
Mediante la conexión a cualquier universidad y algunos institutos de investigación (UNAM, UAM, IPN, ITESM, UDLA, UdeG, CONACYT, CICESE, CIBNOR, CIMAT, MEXNET), es posible el accesar a 2'300,000 *nodos* en todo el mundo, además de tener acceso a 35 universidades públicas y privadas mexicanas y otros organismos del sector público y privado que ya han sido interconectados.

Las siguientes son configuraciones recomendadas que proporciona el ITESM (Instituto de Estudios Superiores Monterrey), siendo de incumbencia solo del usuario el contrato de cualquiera de estas configuraciones.

A lo largo de las láminas se realizan comentarios sobre las ventajas que proporcionan cada una de estas por lo que deben ser tomadas como tales. Estas configuraciones funcionan actualmente en el Campus Estado de México-ITESM.

Opción No. 1

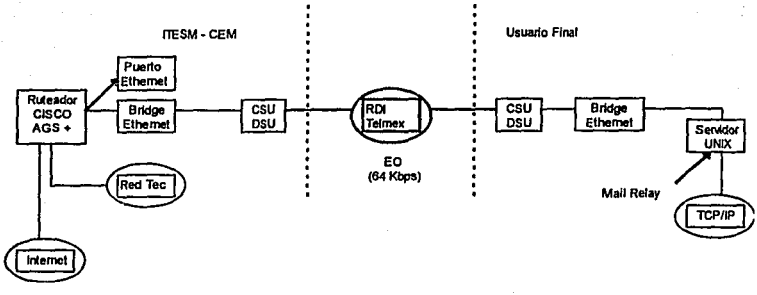
Acceso completo a *Internet*



Recomendaciones:

Toda la *red* del usuario final tiene acceso a *Internet*. Es indispensable que las *direcciones IP* y los nombres de las máquinas estén homologados en el *Network Information Center* (*NIC*) de *Internet*.

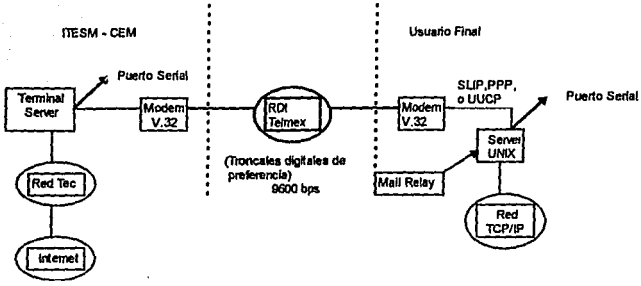
Opción No. 2
Acceso controlado a *Internet*



Recomendaciones:

Únicamente un *servidor UNIX* (con dos interfaces de *red*) tiene acceso completo a *Internet*. Solamente esta máquina debe tener homologada su *dirección IP* y nombre de *host*. El correo electrónico puede pasar a través de dicha máquina en forma transparente, otros servicios de *Internet* requieren tener cuenta en el *servidor UNIX*.

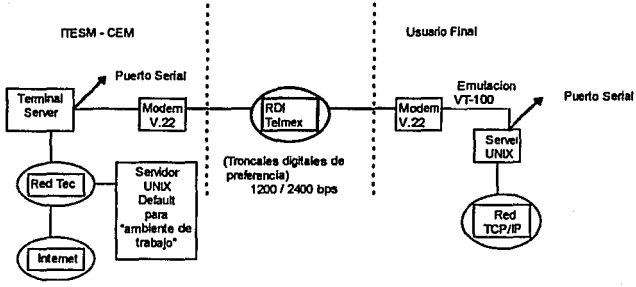
Opción No. 3
Acceso Intermitente a *Internet*



Recomendaciones:

Sólo un usuario del *servidor UNIX* de la derecha puede tener acceso completo a *Internet*, en un momento determinado. Todos los usuarios de la *red* de la derecha pueden mandar y recibir correo *Internet*.

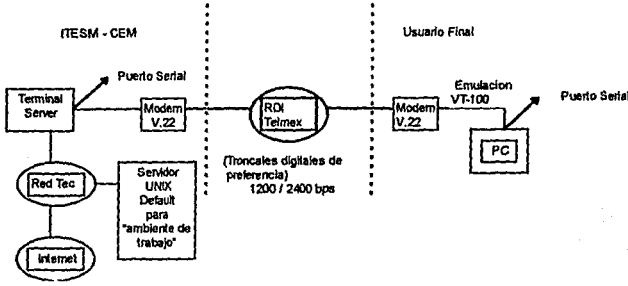
Opción No. 4
Acceso por emulación de terminal



Recomendaciones:

Sólo un usuario del *servidor UNIX* de la derecha puede tener acceso completo a *Internet* en un momento dado (aplicaciones no gráficas). Sólo los usuarios con cuentas en ambos *servidores* pueden mandar y recibir correo.

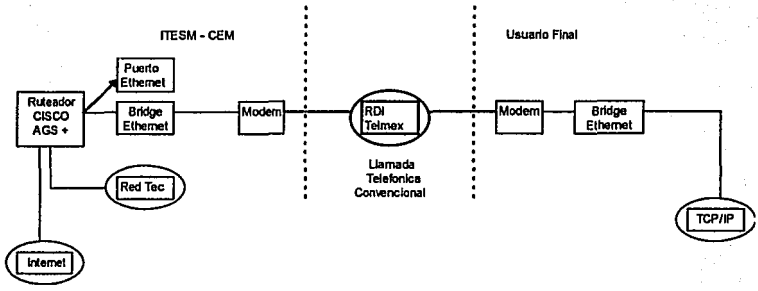
Opción No. 5
Acceso en el D.F. por emulación de terminal



Recomendaciones:

La *PC* actúa como terminal *ASCII* del *servidor UNIX* en el ITESM-CEM. El usuario de la *PC* tiene acceso completo a *Internet* (aplicaciones no gráficas). Limitado por la velocidad de transmisión y la disponibilidad de las líneas telefónicas.

Opción No. 6
Acceso *LAN/LAN* Intermitente



Recomendaciones:

Toda la *red TCP/IP* del Usuario Final tiene acceso completo a *Internet* mientras dura la llamada telefónica. Todas la maquinas del usuario final deben tener *direcciones IP* homologadas. La calidad del servicio solamente esta limitada por la calidad de las líneas telefónicas.

Apéndice V
Resolución de problemas con herramientas de *TCP/IP*

Resolución de problemas con herramientas de *TCP/IP*

Antecedentes:

Las tareas al administrar los servicios de *TCP/IP* se pueden dividir en dos categorías: configuración y resolución de problemas. Las tareas de configuración de alguna forma son predecibles y requieren que se conozca detalladamente la sintaxis de los comandos.

Si un sistema esta optimamente configurado, la razón para cambiarlo deberá ser muy rara. Los procesos de configuración son repetidos cada vez que una nueva versión del *sistema operativo* es instalado, pero aun en esas condiciones los cambios son muy pocos.

En contraste, problemas de la *red* son inesperados. Los problemas de la *red* usualmente son únicos y algunas veces son muy difíciles de resolver. Las solución de los problemas son una parte importante de la estabilidad y la confiabilidad de la *red*.

Para aprovechar un problema, se necesita conocer de forma básica *TCP/IP*. A continuación se daran ideas de aprovechamiento de problemas; no es exactamente una metodología, pero si ideas que se deben tener en mente; la lista no implica un orden específico.

- Aprovecha los problemas metodicamente, toma los registros correctos y anota los resultados, realiza un registro historico del problema en caso de que el problema reaparezca.

- Manten una mente abierta, no asumas que la causa del problema se encuentra en una capa de aplicación si el problema se encuentra en una capa inferior. Algunas gentes asumen que el

problema es siempre la *red*, mientras que también se culpa a los *hosts* remotos de los problemas de la *red*.

- Pon atención a los mensajes de error, algunos mensajes de error son muy vagos, pero frecuentemente contienen información para resolver el problema.

- Muchos de los problemas son causados por errores humanos, es posible prevenir algunos errores si se provee información de la *red* y se entrena acerca del funcionamiento de la misma.

- Toma información de los usuarios, esto *red*uce el número de reportes de problemas.

- No especules del problema con el usuario, guarda tus comentarios para hacerlos con algún colega. Muchas de las especulaciones con los usuarios provocan rumores y que el usuario trate de solucionar el problema provocando un daño menor.

- Usa pocas herramientas para resolver los problemas, las herramientas que se han expuesto son suficientes para diagnosticar cualquier problema. Es posible que el usar una herramienta nueva te tome tiempo dominarla y que el usar una herramienta "vieja" te tome menos tiempo el resolver el problema.

- No niegues lo obvio. La pérdida o caida del cable *Ethernet* es una causa común de problemas en la *red*, checa conectores, cables y switches, regularmente las cosas pequeñas son la causa de grandes problemas.

La causa de muchos problemas tienen un motivo simple, desarrollando una idea clara del problema es posible encontrar una solución de manera pronta. Desafortunadamente esto no siempre es verdadero. Muchas herramientas de diagnóstico estan disponibles con el *sistema operativo*, el adquirir una herramienta de diagnóstico

puede costar unos cientos de dólares, y muy probablemente en la *red Internet* se encuentre *software* libre.

Las herramientas que se explican a continuación son las herramientas de *sistema operativo UNIX*.

Son pocas las razones por las cuales se tiene que adquirir *software* de diagnóstico.

- Un sistema comercial de diagnóstico puede costar cientos de dólares pero se tiene información completa de como usarlo y que utilidad puede tener.
- Muchos administradores prefieren no usar *software* de diagnóstico porque es posible encontrar *software* libre en la *red Internet*.
- Probablemente una *red* grande si necesite un *software* comercial si es que se tiene la misma plataforma y el *software* es a la medida (Como por ejemplo SunNet Manager de Sun Microsystems).

Comandos a Utilizar:

Comandos básicos	Comandos de administración
grep	**ifconfig**
ps	**arp**
	netstat
	ping

En los siguientes apéndices se mostrará el uso e interpretación de los comandos de administración.

Apéndice W

Resolución de problemas con el comando ping

Resolución de problemas con el comando ping

Antecedentes:

Indica si un *host* remoto se encuentra disponible. **ping** sólo despliega estadísticas acerca de los paquetes pérdidos y el tiempo usado para transportarlos.

Procedimiento:

El comando **ping** ayuda a probar si un *host* remoto se encuentra disponible desde el *host* local. Esta simple función es muy usada para probar conexiones en la *red*, independientemente de la aplicación en la cuál se detecto el problema.

El comando **ping** permite deteminar si el problema se encuentra en las capas superiores (de aplicación) o inferiores (conexión de la *red*). Si el **ping** muestra paquetes que viajan por la *red* y regresan el problema muy probablemente resida en las capas superiores y si no se recibe señal de regreso la falla se encontrará en las conexiones del equipo.

Frecuentemente el usuario reporta problemas de conexión cuando realiza un **telnet**, una sesion **ftp** o al enviar un **mail**, al *host* remoto; la herramienta más usada para detectar el problema es el comando ***ping***.

Usa el *hostname* que te provea el usuario y prueba con el **ping** si se tiene la conexión; usa el **ping** de la máquina del usuario para verificar que se tiene conexión al *host* local, si ambas pruebas fuerón exitosas entonces concentra el análisis en la aplicación en especifico que causo el problema.

Muy posiblemente el *host* al que se queria conectar el usuario se encontraba "abajo" cuando se trato de realizar la conexión. Si todo se encuentra correcto y la aplicación continua fallando realiza un análisis detallado de la aplicación con el comando **etherfind** en coordinación con el administrador del *host* remoto.

Si ***ping*** del *host* local es exitoso y el del usuario falla, concentra las pruebas en la configuración del usuario, es posible que se encuentre mal alguna *dirección IP* de la máquina del usuario.

Si el **ping** del *host* local falla y el del usuario falla pon atención a los mensajes de error desplegados estos te pueden dar una ayuda para planear las pruebas subsecuentes.

Algunos de los mensajes mas comunes son:

unknown host

Este mensaje es desplegado cuando el *host* no puede resolver la *dirección IP* asociada al nombre en el archivo ***/etc/hosts***, muy probablemente el *host* no este dado de alta en esta base de datos (***/etc/hosts***) o en realidad la conexión física con la máquina remota se encuentre "abajo" o con fallas. Otra prueba a realizar es, si se conoce la *dirección IP* es posible realizar un **ping** con esta y con esto se elimina el nombre con lo cual la solución es corregir el nombre en la base de datos.

network unreachable

La posible causa es que no existe el *ruteo* necesario para localizar el *host* remoto. Si la *direcciónn IP* fue tecleada con el comando ***ping*** reteclea usando el nombre del *host*, esto elimina la posibilidad de que el nombre sea incorrecto y que dirección asociada sea equivocada. Asegura que las tablas de *ruteo* en el ruteador estén actualizadas o que el *gateway* por default no tenga problemas de *ruteo* con las direcciones que tenga que utilizar.

no answer

El sistema remoto no responde. Muchas *redes* utilizan este tipo de mensaje o alguna de sus versiones. Algunas implementaciones de ***ping*** imprimen el mensaje "100% packet loss", **telnet** imprime el mensaje "Connection timed out" y **sendmail** regresa ***"cannot connect"***. Todos estos mensajes tienen el mismo significado, el sistema local busca al sistema remoto pero no recive respuesta de los paquetes envíados.

Este problema puede tener muchas causas. El *host* remoto puede estar abajo, el *host* local o el remoto puede estar configurado incorrectamente. El *gateway* o el circuito entre el *host* local y el remoto se encuentra "abajo", el *host* remoto puede tener problemas de *ruteo*. Como prueba adicional esposible checar la configuración local usando ***netstat*** y ***ifconfig***.

Procedimiento:

El formato básico del comando ***ping*** *es:*

ping *host {packet size} {count}*

host El nombre del *host* o la *dirección IP* del *host* remoto.

packetsize Define el tamaño en *bytes* de los paquetes de prueba, este campo es usado solo si se usa el campo count, el tamaño por default de los paquetes es de 56 *bytes.*

count El número de paquetes a ser envíados para la prueba, usa este campo para enviar un número de paquetes pequeño, de otra manera el comando ***ping*** continúa enviando paquetes hasta que es interrumpido con *control*-C, si envía demasiados paquetes a través de la *red* solo estas ocupando recursos que no es necesario usar. Finalmente solo 5 paquetes son suficientes para una prueba.

El siguiente es un ejemplo de la manera de usar **ping**, donde marte es un *host* de una *red Ethernet.*

atenea# **ping** *-s marte 56 5*
PING *marte: 56 data bytes*
64 bytes from marte (156.1.14.3) :icmp_seq=0. time=14. ms
64 bytes from marte (156.1.14.3) :icmp_seq=1. time=14. ms
64 bytes from marte (156.1.14.3) :icmp_seq=2. time=13. ms
64 bytes from marte (156.1.14.3) :icmp_seq=3. time=12. ms
64 bytes from marte (156.1.14.3) :icmp_seq=4. time=15. ms

---------- marte Statistics ----------
5 packets transmitted, 5 packets received, 0% packet loss
round-trip (ms) min/avg/max = 12/13/15

La opción *-s* fue incluida por que es una *workstation* Sun, y se incluyen las estadísticas, sin la opción *-s* el comando **ping** solamente contestará "***marte is alive***", en otros sistemas la opción *-s* no se requiere por default las estadísticas se incluyen.

Esta prueba nos muestra una *red* completamente segura puesto que nos informa que nos existierón pérdidas en los paquetes que se enviaron, las estadísticas desplegadas por el comando **ping** puede indicar una *red* con un nivel bajo de problemas.

La llave de las estadísticas son:

- La secuencia (sequence) indica el orden en el cual regresaron los paquetes a la máquina local como son mostrados por el número de secuencia del *protocolo ICMP* (icmp_seq) desplegadas por cada paquete.

- También se muestra el tiempo que tomaron los paquetes en regresar mostrado en milisegundos (time =).

- El porcentaje de los paquetes pérdidos es desplegado como una sumatoria de la salida del comando **ping**.

Si la pérdida de paquetes es alta, el tiempo de respuesta es muy lento, o los paquetes arrivan en orden diferente al que se enviaron esto indica que se tiene una *red* con problemas.

Si se observan estas condiciones en *redes* de área amplia, no hay de que preocuparse. *TCP/IP* fue diseñado para convivir en *redes* seguras, y algunas *redes* de área amplia sufren la pérdida de paquetes. Sin embargo si estos problemas se presentan en una *red* local esto indica problemas.

La mayor parte de los problemas se presentan por los cables, pero si esto no fuera cierto el problema se encuentra en el *hardware*.
Un problema de *Ethernet* puede ser un cable mal terminado o un *segmento* malo una pieza dañada de *hardware* "activo" como un repetidor o un *transceiver*. Checa los terminadores de los cables

primero , esta prueba es muy sencilla de realizar, puesto que estos terminadores son solo resistencias, y estos son causa común de problemas, es recomendable que los extremos del cable no puedan ser accesados por lo usuarios.

Una herramienta que puede ayudar a probar los cables es el reflectometro (*TDR*). Un *TDR* envía una señal y muestra el eco que esta señal produce.

Estos ecos son desplegados en una pequena pantalla, si el cable no esta terminado la señal desplegada brincara hacia la parte de arriba de la pantalla, si el cable es muy corto, la señal sera enviada hacia la parte de abajo de la pantalla, en un cable normal la señal desplegada solamente dara pequeños brincos,con un *TDR* es fácil el detectar un problema de cable.

Los resultados de un simple **ping**, cuando son exitosos pueden ayudar a solucionar directamente las causas del problema. Pero otras herramientas de diagnóstico pueden ayudar a analizar más específicamente y encontrar la causa del problema.

Apéndice X

Resolucion de problemas de acceso a *red* con el comando ifconfig

Resolucion de problemas de acceso a *red* con el comando ifconfig.

Antecedentes:

Los mensajes de error "***no answer***" y "***cannot connect***" indican un problema en las capas inferiores de *red*, si las pruebas preliminares apuntan a este tipo de problemas concentra las pruebas en el *ruteo* y a la interface de red. Usa los comandos **ifconfig**, **netstat** y **arp** para probar la capa de acceso a *red*.

Comandos a utilizar:

Comandos de administración

ifconfig
netstat
arp

Procedimiento:

El comando **ifconfig** checa la configuración de la interface de *red*. Este comando es usado para verificar la configuración del usuario si el usuario no puede "encontrar" otros *hosts* y sin embargo *hosts* remotos si pueden "verlo" y se encuentran en la misma *red*.

Cuando **ifconfig** es usado sin otros argumentos, solo despliega los valores asignados a la interface. por ejemplo si se checa la interface le0 en el *host marte* se obtiene el siguiente reporte:
marte# ***ifconfig*** le0
le0: flags=63<UP,BROADCAST,NOTRAILERS,RUNNING>
inet 156.1.18.4 netmask ffffff00 broadcast 156.1.0.0

El comando **ifconfig** muestra dos lineas en su salida, la primera linea se muestra el nombre de la interface y sus caracteristicas:

UP Esto indica que la interface esta lista para ser usada, si la interface esta abajo (down) es necesario que el administrador "levante" la interface con el comando **ifconfig** (**ifconfig** le0 up).

Recomendación : Si la interface no se "levanta" reemplaza el cable e intentalo de nuevo, si aun falla checa el *hardware.*

RUNNING Esto indica que la interface es operacional. si la interface no se encuentra "running" , el manejador (driver) puede que no se encuentre debidamente instalado.
En este caso es necesario que el administrador cheque todos los pasos de instalación de la interface para encontrar posibles errores o pasos que hayan fallado

La segunda línea de la salida de **ifconfig** muestra la *dirección IP*, la máscara de *subred* (escrita en hexadecimal), y la *dirección broadcast.*

Una máscara de *subred* incorrecta indica cuando el *host* puede no encontrar *hosts* en una *subred* o en otra *red* distante, este comado muestra rápidamente si un número incorrecto de mascara de *subred* es incorrecto.

Un número de *subred* puede causar muchos problemas un tanto extraños, si la parte que indica la *red* en una dirección es incorrecta,

el comando **ping** fallará desplegando un mensaje de error "***no answer***".

En este caso usa el comando **ifconfig** para encontrar una dirección incorrecta.

Sin embargo, si la parte que indica el *host* es la que esta mal, el problema puede ser mas difícil de detectar. En un sistema pequeño como de *PC's* conectadas a otro sistema y nunca se encuentran todos conectados, puede ser que pase mucho tiempo para que se tengan noticias del problema y muy posiblemente la dirección incorrecta se deba solo a un descuido de configuración y para el administrador se vuelva un misterio de problemas intermitentes en la comunicación.

Este tipo de errores puede no ser descubierto por **ifconfig** por que el error es en un *host* remoto, el comando **arp** es usado para este tipo de problemas.

Apéndice Y

Resolución de problemas con el comando arp

Resolución de problemas con el comando arp.

Antecedentes:

Provee información acerca de la traducción de direcciones *Ethernet/IP*, puede ser usado para detectar configuraciones incorrectas de *direcciones IP*.

Comandos a utilizar:

Comandos de administración

arp

Procedimiento:

El comando **arp** es usado para analizar problemas con la traducción de *direcciones IP* a *Ethernet*, el comando **arp** tiene tres opciones para resolver problemas.

-a	Muestra todas las entradas **arp** en la tabla.
-d hostname	Elimina una entrada de la tabla **arp.**
-s hostname ether-adress	Adiciona una nueva entrada a la tabla.

Con estas tres opciones es posible ver el contenido de la tabla **arp**, borra una entrada "problema" y la instala correctamente.

Un claro ejemplo de los errores con la tabla **arp** es cuando un *host* responde con un error a un comando **ftp** o **telnet**, si el problema es intermitente solamente para ciertos *hosts* esto indica que la tabla **arp** esta corrupta. Los problemas en la tabla **arp** son usualmente causados por dos sistemas usando la misma *dirección IP*. Es posible que cuando existan dos direcciones repetidas algunas veces la direccion "correcta" conteste mas rápido que la dirección "incorrecta".

Si se sospecha que existen dos direcciones duplicadas entonces usa el comando ***arp*** con la opcion *-a* para desplegar la resolución de direcciones.

Donde marte y mercurio son *hosts* en una *red Ethernet.*

marte# ***arp*** *-a*
mercurio (156.1.18.1) at 8:0:20:b:4a:71
atenea (156.1.18.2) at 8:0:20:e:aa:40

Si no se tiene este tipo de registro, los primeros tres *bytes* de la *dirección Ethernet* (8:0:20) pueden ayudarte a detectar el problema.

Los primeros tres *bytes* identifican a la compañia que construyó la tarjeta de comunicaciones. Estos prefijos son asignados a cada vendedor que se dedique a la construcción de tarjetas de comunicaciones y son únicos para cada uno de ellos.

Usando esta información se puede identificar el tipo de máquina que se le esta asignando la *dirección IP* "incorrecta".
Para las estaciones de trabajo de SUN se asigno la 8:0:20, supongamos que la dirección que está fallando tiene el prefijo 8:00:38 que nos dice que es una máquina de Bull, y la máquina con la que nos queremos conectar es otra estacion de trabajo SUN entonces el problema se ha reducido a solo cambiar la asignación de *dirección IP*.

La siguiente es una tabla de prefijos de *direcciones Ethernet.*

00:00:0C	Cisco	08:00:0B	Unisys
00:00:0F	NeXT	08:00:10	AT&T
00:00:10	Sytek	08:00:11	Tektronics
00:00:1D	Cabletron	08:00:14	Excelan
00:00:65	Network General	08:00:1A	Data General
00:00:6B	MIPS	08:00:1B	Data General
00:00:77	MIPS	08:00:1E	Apollo
00:00:89	Cayman Systems	08:00:20	Sun
00:00:93	Proteon	08:00:25	CDC
00:00:A2	Wellfleet	08:00:2B	DEC
00:00:A7	NCD	08:00:38	Bull
00:00:A9	Network Systems	08:00:39	Spider Systems
00:00:C0	Western Digital	08:04:6	Sony
00:00:C9	Emulex	08:04:7	Sequent
00:80:2D	Xylogics Annex	08:00:5A	IBM
00:AA:00	Intel	08:00:69	Silicon Graphics
00:DD:00	Ungermann-Bass	08:00:6E	Excelan
00:DD:01	Ungerman-Bass	08:00:86	imagen/QMS
02:07:01	MICOM/Interlan	08:00:87	Xyplex terminal
02:60:8C	3Com	08:00:89	Kinetics
08:00:02	3Com (Bridge)	08:00:8B	Pyramid
08:00:03	ACC	08:00:90	Retix
08:00:05	Symbolics	AA:00:03	DEC
08:00:08	BBN	AA:00:04	DEC
08:00:09	Hewlett-Packard		

Es necesario que se cheque las asignaciones correctas de *direcciones IP* y no los prefijos de los constructores, estos prefijos pueden ayudarte a identificar la fuente de la falla de **arp**, prueba haciendo un **telnet** a la dirección que indica de la *dirección IP* mostrada en el desplegado de **arp**. Si el **telnet** es exitoso entonces configura una *dirección IP* diferente en la base de datos ***/etc/hosts***.

Apéndice Z

Subdivisión de una *red Ethernet*

Subdivisión de una *red Ethernet.*

Antecedentes:

Para reducir el porcentaje de colisiones, se tiene que reducir el tráfico entre *segmentos* de la *red.* Una manera muy simple de hacer esto es el crear multiples *segmentos* de un *segmento* largo. Cada *segmento* debe de tener pocos *hosts* y esto supone poco tráfico, pero como se explicará no es tan sencillo.

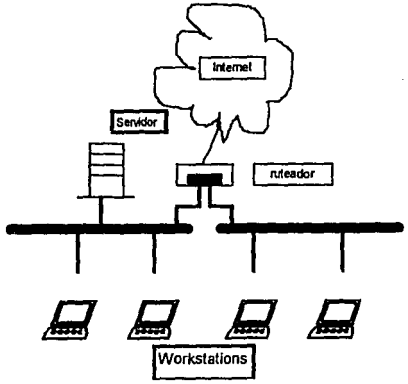
Procedimiento:

Una manera simple y efectiva de subdividir un *segmento* es el de cortar el cable e introducir *ruteadores* o *puentes.* De manera sencilla un *ruteador* es un dispositivo que filtra el tráfico en la *red* basado en las *direcciones IP,* esto crea dos *subredes* y usa las *direcciones de IP* para filtrar el tráfico entre ellas. El otro dispositivo que filtra tráfico en la *red* es un *puente.* Un *puente* filtra el tráfico basandose en la *dirección Ethernet.* La *red* aparenta ser una *subred* pero realmente esta compuesta de dos *subredes.* Generalmente, los *ruteadores* son usados para construir *redes* grandes y los *puentes* son usados para conectar *segmentos.*

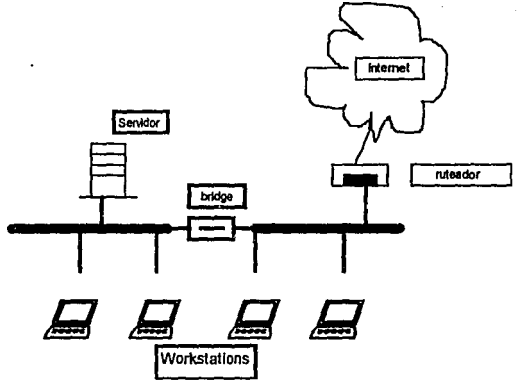
Los *puentes* son mas populares para subdividir *redes Ethernet* por las siguientes razones:

- Los *puentes* son generalmente más baratos que un *ruteador.*
- En ellos no es necesario cambiar las *direcciones IP.* Si se usa un *ruteador* al subdividir la *red* es necesario que algunos *hosts* tengan que cambiar su *dirección IP.*

Subred



Red con *puente*



El el primer ejemplo un *ruteador* que existe es usado, en el segundo caso un *puente* es insertado, Estas ilustraciones ocultan el mayor problema al subdividir una *red*. ¿Dónde se pondrán los servicios

centrales?, si todos los servicios son puestos en un lado de la *red* el tráfico no se verá reducido significativamente.

Antes de insertar un *puente* se tienen que evaluar los servicios que se demandaran, y desarrollar un plan para localizar los servicios que reducirán el flujo de tráfico, es necesario balancear los servicios a ambos lados de la *red*; localiza a los usuarios primarios de servicios a ambos lados del *bridge*, localiza a los usuarios del mismo servicio del mismo lado de la *red*.

Apéndice ZA
Construcción una Pared de Fuego (Firewall)

Antecedentes:

Una *pared de fuego* es un dispositivo o host que ha sido situado entre su red interna y redes exteriores en Internet. Esta pared de fuego no envía broadcast o avisa de rutas, y tiene deshabilitado el re-envío de IP en el kernel. Para salir a alguna red exterior, primero hay que conectarse a este host (pared de fuego) y después continuar a otras redes.

Una pared de fuego se convierte en un elemento esencial en la red y debe mantenerse segura. Es una barrera que protege a la red interna de redes externas. Las restricciones de acceso deben ser reforzadas fuertemente en este host. Hay que considerar imponer lo elementos de seguridad previamente mencionados en este tipo de host.

Comandos básicos:
cp
vi

Para realizar este apéndice es necesario ser el administrador del sistema (quedando bajo su responsabilidad la modificación y el rebooteo del sistema).

Procedimiento:

Para construir una pared de fuego, siga los siguientes pasos :

1. Haga una copia del archivo /etc/init.d/inetinit.

2. Edite este archivo y localice la sección que envía el aviso "Machine is a router :" y haga los siguientes cambios :

- Cambie ndd -set /dev/ip ip_forwarding **1**
  a ndd -set /dev/ip ip_forwarding **0**.

- Cambie /usr/sbin/in.routed **-s**

a /usr/sbin/in.routed **-q**.

- Cambie /usr/sbin/in.rdisc **-r**
  a /usr/sbin/in.rdisc.

3. Vuelva a arrancar el sistema.

El host ya no pasará paquetes de la red interna a las redes externas ni se anunciará a sí mismo como un ruteador.

Apéndice ZB

Resolucion de problemas con el comando netstat

Resolucion de problemas con el comando netstat.

Antecedentes:

Provee una variedad de información, comúnmente es usado para desplegar estadísticas acerca de la interface de *red*, *sockets* de *red* y tablas de *ruteo*.

Comandos a utilizar:

comandos básicos	Comandos de administración
grep	**netstat**

Procedimiento:

El mensaje de error "***network unreachable***" indica claramente un problema de *ruteo*. Si el problema es en la tabla de *ruteo* del *host* local, este es un problema fácil de detectar y resolver, primero usa el comando **netstat** *-nr* y **grep** para buscar un *ruteo* válido.

Este ejemplo checa un *ruteo* específico para la *red* 156.1.14.0
marte# **netstat** *-nr* | **grep** *156.1.14.0*
156.1.14.0 156.1.18.2 UG 0 0 le0

Por ejemplo, un usuario reporta que la "*red* se cayo" por que el comando **telnet** a nic.ddn.mil que es un *host* y al realizar una prueba el comando **ping** le muestra el siguiente resultado:

Donde *marte* es un *host* en una *red Ethernet.*

marte# **ping** *-s nic.ddn.mil 56 2*
PING nic.ddn.mil: 56 data bytes
sendto: Network is unreachable
ping: wrote nic.ddn.mil 64 chars, ret =-1
sendto: Network is unreachable
ping: wrote nic.ddn.mil 64 chars, ret=-1
——— nic.ddn.mil PING Statistics ———
2 packets transmitted, 0 packets received, 100% packet loss

Tomando como referencia el mensaje "***network unreachable***" checa en la table de *ruteo* la dirección del usuario, En el ejemplo anterior podemos ver un *ruteo* a nic.ddn.mil. La *dirección IP* de nic.ddn.mil es 192.112.36.0.

Recuerda que el *ruteo* debe de estar orientado hacia la *red.* Checa el *ruteo* de *red* con 192.112.36.0:

marte# ***netstat*** *-nr* ***### grep*** *'192.112.36.0'*
marte#

Esta prueba muestra que no existe un *ruteo* específico para 192.112.36.0 , si el *ruteo* fué encontrado el comando **grep** lo muestra.

Es necesario recordar que si no se específica un *ruteo* no existe *ruteo* por default a menos de que se defina. Este ejemplo muestra un *ruteo* existoso por default:

marte# **netstat** *-nr* **### grep** *def*
default 128.66.12.1 UG 0 101277 le0

Si el comando **netstat** muestra un *ruteo* correcto, o un *ruteo* por default, el problema no es la tabla de *ruteo.*

Si **netstat** no regresa una tabla esperada, el problema es un *ruteo* local, existen dos maneras de solucionar un problema de *ruteo* local dependiendo si el *ruteo* es dinámico o estático, Si se usa un *ruteo* estatico instala un*ruteo* con el comando **route add** (este comando puede variar dependiendo de la version del *sistema operativo UNIX*, por lo que es necesario consultar los manuales para encontrar la manera de usarlo correctamente).

Si el *ruteo* usado es dinámico asegura que el programa de *ruteo* está corriendo, por ejemplo, asegurate que el demonio **gated** esta "corriendo" con la siguiente instrucción.

marte# **ps** ´**cat** */etc/gated.pid*´

PID	***TT***	***STAT***	***TIME***	***COMMAND***
27711	***?***	***S***	***304:59***	***gated -tep /etc/log/gated.log***

Si el demonio no esta corriendo, inicializa de nuevo el demonio y checa si el demonio no es el problema o si es terminado anormalmente.

GLOSARIO

GLOSARIO

- *.Xauthority* Archivo que crea xdm y que es utilizado, en el control de accesos para almacenamiento y autorización de llaves.

- *.Xdefualts* Archivo utilizado para colocar y almacenar los recursos del cliente.

- *.Xsession* Archivo que almacena y administra una sesión de XDM, se localiza en el directorio de casa del usuario.

- */usr/lib/X11/ncd/configs/* Localización por default para configuración remota de archivos para NCD.

- */usr/lib/X11/ncd/fonts* Localización por default para directorios de tipo de fuente (letra).

- */usr/lib/X11/ncd/rgb.txt* Localización del archivo y/o base de datos de colores.

- *Acceso* En lenguaje de computación, esta palabra se emplea con frecuencia para indicar leer un archivo o escribir en él.

- *Administrador del sistema* La persona responsable del mantenimiento del sistema.

- *ANSI* (American National Standards Institute); Organización que recolecta y pública los estándares en la industria de la computación.

GLOSARIO

● *Archivo* Colección de información relacionada entre sí, que se identifica con un nombre de archivo. El sistema operativo UNIX considera los dispositivos periféricos como archivos, permitiendo que un programa lea o escriba en un dispositivo igual que haría en un archivo normal.

● *Archivo especial* Archivo que representa a un dispositivo. Existen tres tipos de archivos en el sistema UNIX: ordinarios, directorios y especiales (dispositivos).

● *Archivo invisible u oculto* Archivo cuyo nombre empieza con un punto. Reciben este nombre por que el mandato ***ls*** no suele listarlos. La opción ***-a*** de este comando despliega todos los archivos, incluyendo los invisibles.

● *Archivo ordinario* Archivo que sirve para almacenar un programa, texto o datos , a diferencia de los archivos directorio y especial.

● *Argumento* Número, letra o palabra que proporciona información a un programa cuando éste se ejecuta. El argumento de una línea de mandato es lo escrito a continuación del mandato mismo.

● *ARP* (Adress Resolution Protocol); Protocolo de Internet usado para traducir la dirección IP a dirección Ethernet.

● *Arquitectura* Diseño y estructura de el software y/o hardware, de los componentes que consta el sistema.

GLOSARIO

● *Arquitectura de la red* Sistema de cableado, tipo de estación de trabajo y reglas utilizadas en la transmisión de datos entre estaciones de trabajo.

● *ASCII* (American Standard Code for Information Interchange); Es un código que usa siete bits para representar caracteres gráficos (letras, números y signos de puntuación) y de control. El ASCII puede representar tanto texto como programas fuente. Dado que es estándar, se usa con frecuencia para intercambiar información entre computadoras.
Existen extensiones del conjunto de caracteres ASCII que ocupan ocho bits, pero el de siete bits es el más común.

● *Background* Color sólido o dibujo (adornar con dibujos); estos se usan de bajo de caracteres o gráficas en una ventana o menú.

● *Berkeley Software Distribution* Versión de UNIX, desarrollada en la Universidad de California, Berkeley (BSD 4.4).

● *Bit* El bloque de información más pequeño que puede manejar una computadora. Un bit es uno (on: activado) o un cero (off: desactivado).

● *Bitmap* Secuencia de bytes que representan una rejilla de pixeles; usados para figuras o formas de punteros, iconos, dibujos de las ventanas, entre otros.

● *Block* Parte definida de memoria usada, únicamente por la computadora. Un block típicamente consiste de 512 bytes.

● *Bourne shell* Shell de UNIX, usado y definido por los laboratorios Bell.

GLOSARIO

● *Broadcast address* Dirección IP, que hace referencia todos los hosts en una red.

● *Buffer* (área de almacenamiento temporal) Area de memoria que guarda datos hasta que éstos puedan utilizarse. Cuando se escribe en un archivo de disco, el UNIX almacena la información en un buffer de disco hasta tener la suficiente para escribir al disco o hasta que éste se halle listo para recibir la información.

● *Byte* Ocho bits de información. Un byte puede almacenar un carácter.

● *Cables coaxiales* Delgados(thin) y gruesos (thick). Se utilizan con frecuencia en redes de televisión.

● *Cables de par trenzado* (twisted pair): tasas de transmisión reducidas y cobertura de distancias cortas. En general utilizadas como cables telefónicos.

● *Cables de fibra óptica* Transmiten datos por medio de la luz.

● *Cambio (swap)* Pasar un proceso de la memoria a un disco, o viceversa. El cambio de un proceso a un disco permite que otro proceso comience o continúe su ejecución.

● *Carácter alfanumérico* Algún carácter comprendido entre la A y la Z (mayúscula o minúscula) o entre el cero y el nueve.

GLOSARIO

• *Carácter de CONTROL* Un carácter que no es gráfico, es decir, que no es una letra, un número o un signo de puntuación. Se les llama caracteres de CONTROL, porque suelen controlar un dispositivo periférico. Los carácteres de CONTROL, a menudo llamados caracteres no imprimibles, se representan con códigos ASCII menores que 32 (decimal).

• *Carácter especial* Un carácter que no se representa a sí mismo a menos que esté marcado. También son caracteres especiales del shell el asterisco (*) y el signo de interrogación (?).

• *Carácter imprimible* Uno de los caracteres gráficos: letra, número o signo de puntuación.

• *Chip* Pequeña parte de silicon, que transporta el equivalente de un número largo de componentes eléctricos, integrando un circuito.

• *Cliente* Máquina que utiliza los servicios proporcionados por el server (servidor), a fin de levantar sus procesos.

• *CD-ROM* Disco compacto, solo de lectura en memoria; medio de almacenamiento de alta capacidad.

• *Cliente* (Para NCD) Programa de aplicación, para el Sistema de Ventanas X. Más de un cliente puede correr en la computadora central.

• *Código de retorno* Código que indica el estado que devuelve un proceso: éxito (suele ser un cero) o fracaso (un uno).

• *Comando* Instrucción que lleva a cabo la comunicación entre la computadora y una función o tarea especifica.

GLOSARIO

- *Compuerta* (Gateway). Una compuerta es un punto de conexión y traducción entre los diferentes tipos de protocolos.

- *Concatenar* Unir secuencialmente, o extremo con extremo.

- *Conducto (pipe)* Conexión entre dos programas, de modo que la salida estándar de uno se conecta con la entrada estándar del otro.

- *Consola* Cliente local que provee una caja, donde se despliegan mensajes de diagnostico, tiene un menú para poder accesar a otros clientes locales.

- *Control de dispositivo* Programa que controla un dispositivo, como una terminal, una unidad de disco, o una impresora.

- *CPU* Abreviatura de Unidad Central de Procesamiento, está parte de la computadora realiza los cálculos y manipula información.

- *Corchetes* Corchetes ([ ]) o signo de mayor que (>) y menor que (<).

- *Cursor* Pequeño rectángulo o línea luminosa que sale en la pantalla de la computadora e indica dónde aparecerá el próximo carácter.

GLOSARIO

- *CSMA/CD* Carrier Sense Multiple Access/Collision Detect.
  Carrier Sense Detección del tráfico de red.

  Multiple Access Factible entrar por diversos puntos a la red.

  Collision Detect Se detectan posibles colisiones entre envío de paquetes.

- *Demonio* Programa de tipo especial, que inicia cuando se realizan tareas, como envío de correo, impresiones, etc. Proceso en background propio del sistema para optimizar las funciones públicas.

- *Darse de alta o Iniciar por primera vez una sesión en el sistema*
  Obtener acceso a un sistema UNIX, respondiendo en forma correcta a las indicaciones ***login:*** y ***password:***

- *Dataless client* Es una workstation que tiene su propio disco para almacenar operaciones de sistema de archivos.

- *Default* Valor que es utilizado, sino se especifica un valor.

- *df* Comando que reporta espacio en disco ocupado por el sistema de archivos.

- *Dirección ethernet* Dirección de hardware o dirección física, que se define por hardware. Está dirección es un número de 48 bits. Es representado por dígitos hexadecimales y está subdividido en seis campos de 2 dígitos, separados por dos puntos.

- *Dirección internet* Dirección lógica, son 32 bits, dividida en cuatro campos de 8 bits, cada campo de 8 bits es representado por un número decimal entre 0 y 255.

GLOSARIO

- *Dirección IP* Número único con que se identifica a una máquina dentro de una red .

- *Directorio* Archivo directorio; contiene una lista de otros archivos.

- *Directorio de trabajo* Directorio con el que se está asociado en un momento determinado. Los nombres de trayectoria relativos utilizados se basan en este directorio.

- *Directorio raíz* El ancestro de todos los directorios y el comienzo de todos los nombres de trayectoria absolutos.

- *Diskless client* Esta es una workstation que no tiene disco propio, para operar enteramente se debe conectar a una red local.

- *DISPLAY* Variable para configurar el medio ambiente, de la máquina.

- *Dispositivo físico* Dispositivo, como una unidad de disco, separado física y lógicamente de otros dispositivos similares.

- *Dispositivo periférico* Unidad de disco, impresora, terminal, graficador u otra unidad de entrada/salida que puede conectarse al servidor o computadora.

- *du* Comando que despliega el número de bloques usados por cada directorio.

- *Encabezado* Parte de un formato que va en el extremo superior de una página.

GLOSARIO

- *Entrada* Información que requiere un programa desde una terminal u otro archivo.

- *Entrada estándar* Archivo de donde un programa puede recibir datos. A menos que se indique al shell lo contrario, la entrada estándar procederá del teclado de la computadora.

- *EOF(fin de archivo)* Acrónimo de End of File; representa el fin de un archivo.

- *Expresión aritmética* Grupo de números, operadores y paréntesis evaluables. Cuando se evalúa una expresión aritmética, se obtiene como resultado un número.

- *Expresión lógica* Serie de cadenas de caracteres separadas por operadores lógicos (>, >=, =, <=, <=). Después de evaluar una expresión lógica, el resultado puede tomar únicamente dos valores: falso o verdadero.

- *Expresión regular* Cadena (compuesta de letras, números y símbolos especiales) que define a una o más cadenas.

- *Extensión del nombre de archivo* Parte de un nombre de archivo que va después de un punto.

- File system Conjunto de archivos y directorios, ordenados en forma jerárquica de árbol, en el sistema operativo UNIX.

GLOSARIO

- *File System* Jerarquía de directorios y archivos, este es creado en una partición en el disco y consiste de un número de grupos de cilindros, cada uno de los cuales tiene inodos y bloques de datos.

- *fsck* Comando que checa las inconsistencias (tales como inodos sin referencias, bloques extraviados en la lista de bloques libres) en el sistema de archivos y los reporta.

- *ftp* Comando, por medio del cual se comunica, con un host remoto, sólo para efectuar transferencia de archivos.

- *FTP* Servicio de transferencia de archivos.

- *GID* Número que identifica a un grupo. El sistema operativo UNIX asigna este número a un grupo específico de usuarios.

- *GUI* Interfase gráfica de usuario, software que facilita la interacción entre la computadora y el usuario.

- *Home* Una parte del file system donde se guarda la información individual de un usuario.

- *Host* Dispositivo individual en una red.

- *Hostname* Nombre utilizado para identificar únicamente a un dispositivo en una red.

- *Hosts* Una máquina dentro de la red.

- *Icono* Representación gráfica de una aplicación o ventana.

GLOSARIO

- *Identificador de proceso (PID)* Acrónimo de Process Identification (identificación de proceso) que usualmente va precedido de la palabra número. EL sistema UNIX asigna un número PID único a cada proceso en el momento de iniciarlo.

- *Identificador de usuario (UID)* Número asociado al nombre de un usuario.

- *Impresora remota* Impresora conectada a un sistema que es accesible a través de la red.

- *Inodo* Parte de un archivo que contiene información sobre este en el file system. Está información incluye el tipo de archivo, modos de acceso, número identificador del usuario y grupo, número de bytes en el archivo.

- *Inter-redes* (Internetwork). Una interconexión de redes o inter-red (internetwork) es una conexión de dos o mas distintas redes tal que las computadoras en una red se puedan comunicar con las computadoras de otra red.

- *Internet* Colección de LANs y ruteadores usando TCP/IP que funciona como una WAN (Wide Area Network).

- *Kernel* El núcleo del sistema operativo SunOS, que administra los recursos de la computadora.

- *LAN* Local Area Network. Una red de área local, es una red de sistemas de área reducida.

GLOSARIO

- *Local* El sistema al cual están conectadas las terminales en forma directa.

- *Login* Proceso que consigue el acceso al sistema, para iniciar una sesión.

- *Login name* Nombre que se le asigna a un usuario, para controlar sus accesos al sistema.

- *Loopback* Dirección interna, diseñada para pruebas y procesos internos de comunicación en la máquina local.

- *Mbps* Mega bits por segundo.

- *Megaflops* Un millón de operaciones de punto flotante.

- *Método de acceso* Describe como accede un nodo al cableado entre los nodos.

- *MIPS* Millones de Instrucciones por Segundo.

- *Módem* Dispositivo que modula y demodula la transmisión de datos, puede conectarse a una terminal, impresora, servidor, mediante una línea telefónica.

- *MTU* Unidad máxima de transferencia.

- *Multitarea* Permite el acceso a más de un usuario al mismo programa, a un mismo tiempo.

GLOSARIO

- *Multiusuario* Permite que más de un usuario use el sistema al mismo tiempo.

- *NFS* Por default en SunOS provee y distribuye los servicios para poder compartir recursos (archivos).

- *NIC* *Para la presente tesis se tomará como:*
  Network Information Center, definición de NCDware 3.0 Advanced User's Guide for UNIX® Systems.

- *Nombre de trayectoria* Relación de directorios separados por barras diagonales(/). Un nombre de trayectoria sirve para seguir una trayectoria a través de la estructura de archivos y así localizar o identificar un archivo.

- *Nombre de trayectoria absoluta* Trayectoria que comienza con el directorio raíz (/). Un nombre de trayectoria absoluta localiza un archivo sin tener en cuenta el directorio de trabajo.

- *Nombre de trayectoria relativa* Un nombre de trayectoria que parte del directorio actual y no de la raíz, como el nombre de trayectoria absoluto.

- *Núcleo (kernel)* El centro del sistema UNIX. La parte del sistema operativo que asigna recursos y controla procesos.

- *Número hexadecimal* Número de base 16; compuesto por los dígitos hexadecimales del cero al nueve y de la A a la F.

- *Path* Lista de nombres de directorios, donde puede buscar y ejecutar algún comando.

GLOSARIO

- *Path name* Lista de nombres de directorios , separados por slashes (/), que especifican la localización particular de un archivo.

- *Permiso de acceso* Permiso para leer de un archivo, escribir en él o ejecutarlo.

- *Proceso* Medio por el cual UNIX ejecuta un programa.

- *Proceso preferente* Cuando un programa se ejecuta de manera preferente, está ligado al servidor. A diferencia de un proceso subordinado, en éste debe esperarse a que termine su ejecución para poder dar otro mandato al Shell.

- *Proceso subordinado* Proceso que no se ejecuta de forma preferente, también se denomina proceso separado. Un proceso subordinado se inicia con una línea de mandato que termina con un signo &. No hay que esperar a que termine el proceso para digitar otros mandatos de Shell.

- *Programa Shell* Proceso compuesto por mandatos Shell.

- *Protocolo* Conjunto de reglas para indicar el modo de comunicación a través de la red. Reglas y procedimientos utilizados en la red para establecer comunicaciones entre los nodos. Convenciones acordadas entre los diferentes nodos para el envío de la información.

- *ps* Comando que reporta el estado de los procesos, que corren en el sistema.

- *PostScript* Lenguaje de Programación.

GLOSARIO

- *Puentes* (Bridge). Un puente permite a las redes interconectarse con otras; y copian paquetes de una red a otra.

- *rcp* Comando para realizar copias remotas, entre dos estaciones de trabajo, y pueden ser tanto archivos como directorios.

- *Red* Sistemas dispuestos a comunicarse mediante la combinación de hardware y software.

- *Redes extensas WAN* (Wide Area Network). Se extienden a superficies planetarias y conectan máquinas en diversos países.

- *Redes locales LAN* (Local Area Network). Redes pequeñas generalmente ubicadas en un solo lugar físico o a lo largo de una extensión reducida.

- *Redes metropolitanas MAN* (Metropolitan Area Network). Conjunto de redes locales conectadas entre si dentro de una extensión más amplia que una LAN.

- *Redireccionamiento* Proceso en el que se dirige la entrada estándar de un programa para que proceda de un archivo que no sea el teclado de la terminal, o proceso en el que se redirecciona la salida estándar hacia un archivo en lugar de la pantalla de la terminal.

- *Remoto* Sistema el cual, es accesado a través de la red.

- *Repetidores* (Repeater). Un repetidor sólo copia bits de una red a la otra.

GLOSARIO

- *rsh* Comando que realiza la ejecución de algún comando , en el servidor remoto, sin necesidad de establecer una sesión remota.

- *Ruteadores* (Routers). Un ruteador mueve paquetes de una red a otra, toma decisiones del camino que seguirá la información.

- *Salida* Información que un programa envía a la terminal o a otro archivo.

- *Servidor* Máquina que proporciona todos los servicios a una red.

- *Señal* Mensaje breve que UNIX envía a un proceso, sin tener en cuenta la entrada estándar del proceso.

- *Servidor de X (X server)* Software que corre en la estación de trabajo o en la terminal de despliegue.

- *Sesión* Secuencia de eventos que ocurren desde que se comienza a usar u programa como el editor hasta que se termina, o desde que el usuario se da de alta hasta que termina.

- *Shell* Interprete de comandos UNIX, el shell interactua entre el usuario y el *kernel.*

- *Shell de entrada* el shell que se utiliza al iniciar una sesión por primera vez. Este shell puede generar otros procesos que ejecutan otros shells.

- *Sistema de ventanas X* Sistema de Ventanas en red, basado en el *Protocolo X*, para terminales de mapeo de bits.

GLOSARIO

● *Sistema operativo* Es un programa que maneja todos los recursos de la computadora, incluyendo la unidad de control de procesos, memoria y periféricos de almacenamiento externo como discos y cintas. El sistema operativo también controla el uso de terminales e impresoras.

● *SMTP* Servicio de correo electrónico.

● *Standalone* Es un sistema de completo con monitor, memoria y disco.

● *Subdirectorio* Directorio localizado dentro de otro directorio.

● *Subred* Extensión del esquema de direccionamiento internet que permite que un sitio con redes físicas múltiples, sea accesado desde internet usando una sola dirección de red IP.

● *Superusuario* Usuario privilegiado que tiene acceso a cualquier cosa en el sistema. El administrador del sistema debe estar capacitado para convertirse en superusuario para habilitar nuevas cuentas, cambiar contraseñas y realizar otras funciones administrativas.

● *Telnet* Comando que inicia la conexión a otro host como terminal virtual, o Servicio de emulación de terminal.

● *Termcap* Abreviatura de terminal capability (capacidad de la terminal). El archivo *termcap* contiene una lista de diferentes tipos de terminales y sus características. Los programas orientados a manejo de pantalla, como el editor *vi*, utilizan este archivo.

GLOSARIO

- *Terminal X* Unidad que despliega aplicaciones de múltiples *hosts, (servidores de aplicaciones)* en una red usando un sólo sistema de ventanas.

- *Topología de la red* Esquema físico del sistema de cableado entre los nodos de la red.

Principales topologías: Formas líneales, de anillo o circulares y en forma de estrella.

- *Username* Nombre válido para accesar al sistema.

- *Variable* Nombre y valor asociados que se usan en un programa como, por ejemplo, el shell.

- *Variables de ambiente* Nombres, que define el sistema y/o el usuario para referencia a servicios o información que el sistema necesita para operar.

- *Xlib* Colección de subrutinas usadas por las aplicaciones para comunicarse con el servidor de X. Xlib comprende todas las utilerías implementadas para la versión de X a la fecha, modificadas y actualizadas.

Conclusiones:

La necesidad de trabajar en equipo con objetivos comunes, temas diversos y de moverse hacia la industrialización y automatización apoyando a las empresas a ser competitivas, así como captar las necesidad de las empresas, para tratar de aportar soluciones que le ayuden a elevar su nivel de competencia y desarrollo fue uno de los puntos más importantes que se consideraron, durante el desarrollo de la presente tesis.

Se planteó básicamente la problemática derivada de la falta de conocimientos acerca de la administración del sistema operativo Solaris 2.X, del protocolo de comunicaciones TCP/IP y del protocolo X, así mismo se plantearon varias hipótesis tendientes a demostrar que mediante la conjunción de los tres pueden ser solucionados algunos problemas específicos dentro de una red como lo podrían ser la subutilización de recursos -impresión, archivos, transferencias, aplicaciones gráficas-, la tolerancia a fallas, la seguridad de la información.

Recordemos que las redes se han desarrollado para conectar las máquinas (equipo de cómputo) existentes, para aprovechar el bajo costo y alto rendimiento de los microprocesadores, para lo cuál se requieren conocimientos y experiencia, que difícilmente podemos encontrar en un libro.

Haciendo una analogía entre una red y la administración de servicios de red, en los cuales la primera se diseña como una serie de capas de protocolos, en donde cada una de las capas es responsable de algún aspecto operativo de la red, y la otra involucra: conocimientos y practica a diferentes niveles, que van de lo técnico a lo administrativo, y que en conjunto hacen posible una buena administración de servicios de red.

Durante el desarrollo de la tesis pudieron ser comprobadas las siguientes hipótesis:

"A través de TCP\IP se pueden compartir periféricos, servicios de impresión e información desde cualquiera de los nodos dentro de una red."

Esta hipótesis pudo ser validada en el desarrollo del tema nueve, ya que ahí se habla acerca de aplicaciones de TCP/IP, que permiten entre otras cosas compartir información en los casos específicos de NFS y Telnet, en el apéndice J se habla acerca de los respaldos y de cómo es posible utilizar dispositivos conectados a máquinas remotas en el caso de que localmente no se cuente con tales recursos a través de aplicaciones de TCP/IP, y para el caso de impresoras los apéndices S y T muestran un caso práctico para la instalación de una impresora remota a través de TCP/IP.

"Si se utiliza el protocolo X, entonces será posible tener varias aplicaciones desplegándose simultaneamente en una terminal X."

Toda vez que se tuvo la oportunidad de hablar del protocolo X de comunicaciones se hizo mención de que una de sus principales características era la posibilidad de realizar el despliegue de diferentes aplicaciones en una misma terminal, y esto pudo ser comprobado en los temas 5 y 10.

"Se puede realizar la administración de una red bajo Solaris 2.X desde cualquier punto de la misma, si se auxilia de los protocolos TCP/IP o X."

"Si se implementa NFS en la red, se podrá disminuir la duplicidad de información."

Esto es una consecuencia de la filosofía de trabajo del NFS, ya que esta aplicación fue diseñada pensando en un entorno de trabajo de red, en el cual pudieran compartirse archivos en toda la red de manera que puedan ser accesados como parte del sistema local sin formar parte de él en realidad y de esta manera cualquier usuario dentro de la red pueda tener acceso a la misma información.

Si se reconfigura el Kernel de Solaris 2.X en base a las necesidades de trabajo de los usuarios, entonces puede mejorarse el desempeño del sistema.

Si se configuran equipos como servidores de impresión, entonces se pueden reducir los costos de operación dentro de las empresas, al utilizar de forma más eficiente el equipo con que se cuenta.

La presente tesis se concluyo reafirmando todas aquellas hipótesis que en un inicio nos llevarón a desarrollarla, conjugando los conocimientos adquiridos en la universidad, en el campo laboral, libros, revistas, investigación, y sobre todo la experiencia de cada uno de nosostros.

Si bien la administración básica de los servicios de red en Solaris 2.X a través de TCP/IP, no es sencilla, tiene algunos puntos a su favor, como la potencia que ofrece el sistema operativo UNIX, que es un sistema multiusuario, con capacidad de simular multiprocesamiento, empleo dinámico de memoria, permite comunicación entre procesos, garantizar un alto grado de portabilidad, entre muchas otras características importantes. Permitiendo que la administración sea menos compleja, y eficiente, ante las necesidades de los usuarios y empresas, que actualmente en el área de informática se habla de diferentes sistemas operativos, diversidad en equipos, lo que nos hace pensar que debemos abrir un espacio en nuestra facultad para el mundo de UNIX, pensando que existen empresas donde convergen dos partes muy importantes del mundo de la informática: El primero el sistema operativo MS-DOS y el segundo, no menos importante y que pretende sustituir al primero en un plazo no muy largo, el sistema

operativo UNIX. Bajo estas circunstancias y considerando algunos comentarios: "El profesionista tarda más de un año en producir para la empresa, antes de esto es una inversión. ". Doctor Francisco José Enciso Aguilar, Subdirector de Administración de Riesgos de BITAL, Banco Internacional.

De está manera pretendemos que la presente tesis sea algo más que un simple libro de estante, con hojas amarillentas, más bien que sea una base, para aquellas generaciones que nos siguen, logrando reducir el tiempo de preparación y una rápida incorporación en el mundo laboral.

Bibliografía:

1. Bach, J. Maurice.
   The design of the UNIX operating system.
   Estados Unidos, Prentice Hall, 1986.

2. Morgan, Rachel.
   Introducción al UNIX sistema V.
   México, McGraw Hill, 1990.

3. Manual de instalación de Sun Microsystens.

4. G. Kochan, Stephen y H.Wood, Patrick
   UNIX networking.
   Estados Unidos, Hayden Books, 1989.

5. Quercia, Valerie y O'Reilly, Tim.
   X windows system user's guide.
   Estados Unidos, O'Reilly & Associates, Inc.

6. Manual del curso de Sun Microsystems "System Administration LL"

7. Manual del curso de Sun Microsystems "Advanced System Administration"

8. Manual de Instalación de Sun Microsystems UNIX Networking

9. O'Reilly, Tim.
   TCP/IP
   Estados Unidos, O'Reilly& Associates, Inc.

10. Rifflet
    Comunicaciones en UNIX
    Editorial McGraw-Hill

11. Getting Started with NCD X terminals
    NCD Ware Ver 3.0 part number 93 00 189 January, 1993

NCD
Network Computing Devices,. Inc.
350 North Bernardo Avenue Mountain View Cal. 94043

12. NCD Ware 3.0
Ver 3.0 Reference Manual part number 93 00 192 January 1993
NCD
Network Computing Devices,. Inc.
350 North Bernardo Avenue Mountain View Cal. 94043

13. NCD Ware 3.0
Ver 3.0 Reference Manual part number 93 00 991 January 1993
NCD
Network Computing Devices,. Inc.
350 North Bernardo Avenue Mountain View Cal. 94043

14. X Window System Program
Nabalyoti Barakakati
De. SAMS
A division off Mamillan Co.
Computer Publice Publishing
117 North College, Carnel, Indiana

15. X Window System
C libraries and protocol reference
James, Gettis. Ron, Newman. Robert, W. Sheifler
Digital Press.

16. Vol. 0 X Protocol Reference manual
Robert W. Sheifler
O'Reilly Associates, Inc.

17. Artículos de la revista RED
Copias ilegibles (Autor desconocido)

Seguridad en la información

Herramientas de ingeniería de software asistidas por

computadora

Programación orietada a objetos

UNIX vs. Windows NT
Servicios de impresión compartida