12

UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO

FACULTAD DE INGENIERIA

"APLICACIONES DESARROLLADAS EN EL SISTEMA OPERATIVO LINUX PARA SOLUCIONES DE ALGUNOS PROBLEMAS DE REDES DE COMPUTADORAS"

T E S I S

QUE PARA OBTENER EL TITULO DE:

INGENIERO ELECTRICO - ELECTRONICO
(ELECTRONICA PARA COMUNICACIONES)

P R E S E N T A

ALEJANDRO VELAZQUEZ MENA

DIRECTOR DE TESIS: DR. JESUS SAVAGE CARMONA

MEXICO, D. F. 2001

Agradecimientos

A mi madre por todo el amor y apoyo que me ha dado en todos estos años.
A mi madre por su compresión y sobretodo porque es mi madre.

A mis hermanos (Maribel, Delia y José Miguel) que siempre están ahí para escucharme y apoyarme y darme su compresión aunque a veces no lo demuestro.

A mi padre por enseñarme valores y demostrarme otra forma de querer a alguien.

A Eduardo García Mena por enseñarse valores y una forma de ver la vida muy diferente.

A María de los Ángeles por ayudarme a ser como soy.

A Guillermo Vega por sus consejos y apoyo.

A la UNAM y sus maestros por sus enseñanzas.

A Laura porque siempre estuvo ahí.

INDICE GENERAL

INTRODUCCIÓN

1.1 Antecedentes

El número de computadoras conectadas a Internet se ha duplicado cada año desde hace 10 años. Todo indica que este crecimiento continuará durante muchos años más. Al aumentar el interes en Internet, también ha aumentado el interes en el sistema operativo unix, especialmente porque es unos de los favoritos para los "gateways" de internet, en las computadoras de investigación y para las plataformas educativas, considerando los nuevos sistemas operativos libres que contiene todas las ventajas de los sistemas operativos unix tradicionales ademas de obtener su codigo fuente para conocer realmente como trabajar el comando, el servicio o la aplicación dentro de este sistema operativo. Cuando se hablan de soluciones críticas, en cuanto a nivel sistema operativo no se duda en usar UNIX.

1.2 Planteamiento del Problema

Uno de los problemas actuales que tiene la División de Ingeniería Eléctrica (DIE) de la Facultad de Ingeniería (FI), es la escasez de direcciones IP (Internet protocol). Actualmente (2001) se le ha asignado 254 direcciones IP y el número de computadoras que tiene que mas de 400; sé necesita, en general un gateway con doble acceso, el sistema operativo Linux es una propuesta inicial, para ello se debe conocer previamente la instalación y requerimientos posteriores del sistema operativo, administrar los recursos del sistema operativo, aplicaciones y usuarios. Por otra parte hay que desarrollar y tener la solución para los problemas en cuanto a aplicaciones avanzadas en las redes de computadoras.

1.3 Objetivos

Aprendera a manejar y administrar un sistema operativo Linux.

Analizar, diseñar y construir un servidor que comparta archivos a computadoras windows desde una computadora Linux.

Conocer el protocolo TCP/IP (Transmission Control Protocol) a nivel de paquetes para el análisis, diseño construcción de un "firewall" con doble acceso para incrementar la seguridad en las computadoras de la red local.

1.4 Organización

Todo esto es la base del presente trabajo, en el cual está dividido en cuatro capítulos. El primero se refiere a instalación y/o administración del sistema paso a paso así como los requerimientos del mismo.

Linux es un sistema Unix que tiene una relación muy fuerte con el protocolo de comunicación TCP/IP, en este capitulo hablamos de archivos de configuración y comandos comunes en UNIX.

En el segundo capítulo se desarrollan las tareas más comunes al administrar el sistema operativo como:

Administrar usuarios, manejo de discos, manejo de memoria, respaldo, recuperación de datos, etcetera.

El tercer capítulo se desarrollan aplicaciones para las futuras implantaciones a los problemas antes mencionados, esto se puede desarrollar con el conocimiento previo de los capítulos anteriores.

El cuarto capítulo dará los resultados correspondientes a los temas expuestos y a las pruebas en cuanto al funcionamiento, rendimiento, etc.

Al final de este trabajo se hará un manual de prácticas sobre los temas desarrollados, con el fin de apoyar a la materia de redes de computadoras y también tenerlo disponible en el WEB para uso público e incrementar los conocimientos sobre del Sistema Operativo Linux.

Linux es un clon libre del sistema operativo Unix, que corre sobre computadoras personales (PC's) basadas en procesadores Intel 80386, 80486 y Pentium, además también opera en estaciones de trabajo con procesadores Sparc, Alpha y M68K.

Linux soporta un amplio rango de software tal como TeX, X Window, Matlab, Java, Corel WordPerfect, Scilab, SSH, GNOME, KDE, StarOffice, Adobe Reader, Netscape Navigator y Communicator, XEmacs, Wabi, Perl, Python, Siag, Tcl/Tk, el compilador GNU C/C++ y TCP/IP, además que compañías comerciales ya están liberando producto para Linux ejemplo: Sybase, Informix, Oracle, TIS, Corel WordPerfect, etcétera. Es muy versátil, distribuido libremente en los términos del GNU Licencia Pública General.

95% de los usuarios de Linux lo manejan exactamente igual que si estuvieran usando otro sistema Unix.

LINUX versus UNIX

Linux no tiene conexión con la marca registrada Unix, es una marca registrada de X/open.

Unix es uno de los más populares sistemas operativos en el mundo porque tiene una amplia base de soporte y distribución. Este fue desarrollado como un sistema multitarea para minicomputadoras y *mainframes* a mediados de los años 70, desde entonces ha crecido convirtiéndose en uno de los sistemas operativos más usados, a pesar de su confusa interfaz gráfica de usuario (GUI) y falta de estandarización central.

Unix es un sistema operativo multiusuario y multitarea. Esto significa que varias personas pueden estar usando una computadora al mismo tiempo, corriendo diferentes aplicaciones esto lo diferencia de MSDOS, en el cual una persona puede estar corriendo el sistema a la vez.

Bajo Unix, los usuarios se identifican con el sistema, ellos tienen un registro de entrada, el cual requiere dos pasos: meter un login (el nombre con el cual el sistema le identifica), y meter el password, el cual es la llave secreta personal para registrarte en la cuenta, nadie puede entrar al sistema con el nombre de usuario.

Cada sistema Unix tiene un nombre asignado (hostname), este nombre da a la computadora una identidad, define su carácter, clase y encanto. El nombre es usado para identificar la computadora en una red.

Existen versiones de Unix para algunos sistemas, que van desde computadoras personales hasta super computadoras. La mayoría de las versiones de Unix para computadoras personales son bastante caras e incómodas.

Linux es libre, esto resuelve la parte costosa, es además muy poderoso, fácil de instalar y mantener por una persona, lo que representa una ventaja.

Linux es una versión de Unix, distribuida libremente, desarrollada primero por Linux Tovalds para la Universidad de Helsinki en Finlandia. Fue desarrollado con la ayuda de algunos programadores de Unix y expertos que cruzan la Internet, dando la habilidad para desarrollar y cargar el sistema para cualquier persona, con suficiente conocimiento e iniciativa para escribir en forma rutinaria sobre el kernel de Unix.

Unix y sus clones han sido percibidos como un largo recurso, que requiere muchos discos. Linux es pequeño, rápido y flexible.
Algunas características importantes de Linux que lo hacen único son:

a) Realiza multitareas y soporta 32 bits.

Linux, como todas las otras versiones de Unix, es un verdadero sistema multitarea, habilitan a múltiples usuarios para correr algunos programas sobre el mismo sistema a la vez. El desempeño de un sistema 486 a 55 MHz corriendo Linux es comparable con algunas pequeñas y medianas estaciones de trabajo corriendo propiamente versiones de Unix. Linux es además un completo sistema operativo de 32 bits, utiliza las características del modo de protección especial para los procesadores Intel 80386 y posteriores.

b) Utiliza el sistema X Window.

El sistema X Window es el sistema gráfico estándar para computadoras Unix. Una versión completa del sistema X Window, conocido como Xfree86, está disponible para Linux. El sistema X Window es una interfaz gráfica de usuario (GUI) muy poderosa soporta algunas aplicaciones; por ejemplo, se puede tener múltiples sesiones con el registro de entrada de

diferentes ventanas sobre la pantalla al mismo tiempo. Actualmente la nuevas interfases gráficas de usuarios (GUI) en este sistema operativo es *GNOME* (GNU Network Object Model Environment) y KDE (k desktop environment).

c) Soporta TCP/IP (Transmission Control Protocol/Internet Protocol)

Este es el conjunto de protocolos que liga a millones de computadoras de universidades y negocios hacia la red mundial conocida como Internet. Con una conexión Ethernet, se puede tener acceso a la Internet o la red de área local desde el sistema Linux. Usando SLIP (Serial Line Internet Protocol) ó PPP (Point to Point Protocol) se puede tener acceso a la Internet a través de las líneas telefónicas con un modem.

d) Memoria virtual y librerías compartidas.

Linux puede usar una porción del disco duro como memoria virtual, expandiendo la cantidad total disponible de RAM. Linux además utiliza librerías compartidas, permitiendo a los programas que usen subrutinas estándar, encuentran el código para estas subrutinas en las librerías al momento de correr. Esto salva una gran cantidad de memoria RAM del sistema, porque cada aplicación no almacena su propia copia de estas rutinas comunes. Linux además utiliza toda la memoria RAM del sistema, sin límites de memoria o segmentación, a través del uso de un manejador de memoria virtual.

e) El kernel de Linux no usa código desde cualquier otra fuente propia.

Mucho del software disponible para Linux es libre. De hecho, un gran número de utilidades en Linux son desarrolladas por los proyectos GNU para la Fundación de Software libre en Cambridge, Massachusetts. Sin embargo, entusiastas de Linux, hackers, programadores, y recientemente compañías comerciales desde todo el mundo, han contribuido para el crecimiento del software de Linux.

f) Linux soporta (casi) todas las características de las versiones comerciales de UNIX.

Algunas de las características que se encuentran en Linux podrían no estar disponibles en otros sistemas propietarios de Unix.

g) Soporta software GNU.

Linux soporta un amplio rango de software libre del proyecto GNU, incluyendo utilidades como son el compilador GNU C y C++, GNOME, gawk, groff y más. Muchas de las utilidades esenciales de Linux son software de GNU.

h)Linux es cerradamente compatible con el IEEE POSIX.1 estándar.

Linux ha sido desarrollado teniendo en mente la portabilidad de software, así soporta muchas de las características importantes de otros estándares de Unix.

i)El sistema Linux corre exclusivamente en modo de 32 bits.

Así, esta por encima de un limitador entero de 16 bits en MSDOS. Linux tiene construido soporte para redes, multitarea, y otras características. Se considera "nueva tecnología" a sistemas tales como Windows NT. De hecho, Unix (y ahora Linux) implementaron esta "nueva tecnología" desde hace 15 años.

j)Linux es pequeño.

Linux usa menos memoria y recursos o espacio de disco que algunos sistemas MSDOS o Microsoft Windows. Esto incluye largas aplicaciones (tales como Microsoft word o lotus 1-2-3).

Linux está en un constante estado de desarrollo.

Es difícil continuar con las revisiones que llegan sobre un elemento principal diario en los sitios ftp de la Internet.

Linux es más barato de obtener que la mayoría de sistemas Unix y sus clones disponibles comercialmente.

Si se tiene acceso a la Internet, el único precio que se paga por instalar Linux es el tiempo. Linux está disponible libremente en la Internet.

La ventaja más importante de usar Linux es que se trabaja con el kernel real. Es decir, todo el código fuente del kernel es disponible para Linux, y se tiene la habilidad para modificarlo y cubrir sus necesidades. Trabajar el código fuente del kernel es una experiencia educacional.

k) Requerimientos de hardware

La diferencia de otras versiones de Unix para computadora personal (PC), Linux es muy pequeño, se puede correr un sistema desde un solo disco de alta densidad 5.25. Sin embargo, para correr un sistema de Linux completo, se tienen otras necesidades de Hardware.

Linux por su misma naturaleza es continuamente expandido, y más características se le anexan cada día. Sin embargo, la compatibilidad del hardware es limitado al que tienen los desarrolladores.

Afortunadamente, hay algunos drivers genéricos de disco duro IDE para Linux. Estos drivers genéricos deberían trabajar con todos los discos duros IDE y sus adaptadores. La mayoría de los drivers tipo interno son soportados, pero drivers tipo externo que corren conectándose al puerto paralelo de la impresora generalmente no son soportados.

La siguiente es una guía de algunos requerimientos para Linux. No se tiene que seguir exactamente, pero le dará una idea general de que se requiere:

Un sistema basado en procesador Intel 80386, 80486, Pentium, Pentium II, Pentium III, AMD o Ciryx.

No se necesita un coprocesador matemático, pero es muy recomendable que se tenga uno. Si se tiene un chip 80386, 80486 los coprocesadores matemáticos están disponibles en forma separada, y estos se instalan en un socket en la tarjeta madre (motherboard) de la computadora. Si se tiene un procesador 80486, el coprocesador ya lo trae incluido (excepto el 80486SX, el cual es un chip 486 con el coprocesador deshabilitado).

Si no se tiene coprocesador matemático, el kernel de Linux emula operaciones de punto flotante. Si se tiene uno, las operaciones con punto flotante son manejadas por el hardware, lo cual aumenta considerablemente la velocidad de algunas aplicaciones.

Los procesadores con los que Linux trabaja sin ningún problema van desde el 80386SX hasta el procesador Pentium.

La arquitectura puede ser ISA, EISA o Local bus.

El tipo de bus especifica como el CPU se comunica con el hardware y es una característica de la tarjeta madre. La arquitectura más común es la ISA.

Si la computadora usa Local bus, esta es ampliamente recomendada por que ésta con el VESA local bus estándar (la mayoría de los sistemas de bus local lo hace). Pentium con PCI bus vídeo tampoco tienen ningún problema.

El bus EISA es más reciente que el bus ISA, es más rápido en algunas computadoras. La arquitectura de bus local es la más rápida de las tres, porque ésta permite al CPU comunicarse directamente a vídeo y a los adaptadores de drivers.

La Arquitectura Microcanal (MCA) y la línea IBM PS/2 ahora son soportadas.

Un mínimo de 4 MegaBytes (MB) de RAM en su computadora.

Técnicamente, Linux es capaz de correr en sistemas con solo 2MB de RAM; sin embargo, algunas distribuciones de Linux requieren 4MB de RAM para su instalación.

Memoria significa rapidez, si se tiene más memoria física el sistema hará menos intercambios a disco (swap, intercambio), 8MB serían más que suficientes para la mayoría de las aplicaciones, más de 8MB de RAM definitivamente aumenta la velocidad de algunas aplicaciones. De hecho, si se quiere utilizar el sistema X window, 8MB son requeridos como mínimo y para GNOME mínimo 24MB.

Linux usa los primeros 640KB para texto de kernel, datos de kernel y para el búfer cache. La tarjeta madre utiliza los 384KB restantes para el conjunto de chips. También está el proceso que inicia el sistema y posiblemente otros *daemons*. Entonces, mientras compila, se necesitan como 2.57 a 770KB. Así, si no se tiene suficiente memoria real se tiene que recurrir a la paginación desde disco (swap).

Un controlador de disco duro estándar compatible.

Este incluye MFM, RLL, ESDI y controladores IDE. Algunos controladores SCSI también son soportados. Estos términos especifican la manera de comunicarse con el disco duro a través de tarjetas controladoras. La mayoría de tarjetas controladoras son IDE o SCSI.

Requerimientos de espacio de disco.

Se necesita un disco duro con suficiente espacio disponible para instalar Linux. La cantidad de espacio requerida depende de la cantidad de software que se instale, y el espacio que se necesite para guardar lo que se requiera.

Si se instala lo mínimo, menos de 50MB son suficientes. Se puede instalar un número opcional de paquetes de software, incluyendo el sistema X window, con lo que quizás 200MB o más serán requeridos. Además, probablemente se necesite una cantidad aparte de espacio en disco duro para la partición de swap, la cual es usada para el manejo de la memoria virtual.

En general, se deberá considerar 200MB de espacio de disco para uso del sistema, 16MB más para un espacio de swap y una cantidad extra para los programas y software personal. El espacio de swap es un área sobre el disco que Linux utiliza para almacenar imágenes de programas en proceso cuando la memoria principal se encuentra muy saturada.

Linux soporta la mayoría de controladores de disco duro que son compatibles con un controlador de disco Western Digital WD1003 MFM. Este controlador fue el más común para PC-AT, la mayoría de AT MFM, RLL, ESDI y IDE son también aceptados. Los MFM, IDE y la mayoría de dispositivos SCSI trabajan sin ningún problema.

Requerimientos especiales para el sistema X window:
Con 4MB de RAM el X window corre demasiado lento, se deberá tener mínimo 8MB de RAM para compilar y correr programas en X window. Se necesitan otros 6MB o 10MB de espacio de disco para el compilador GCC, aparte de lo ya mencionado para el sistema X window.

Tabla de comparación de Linux

Procesador	RAM	No. de Clientes	Tipo de Carga
386	8MB	1	Experimento con Linux/Samba; Firewall
386 con DMA y discos SCSI	16MB	1-8	Procesador de palabras; Firewall; Nivel de Datos para el usuario
486	8MB	1-4	Procesador de palabras; Firewall; Nivel de Datos para el usuario
486 con DMA y discos SCSI	16MB	1-10	Procesador de palabras; Firewall; Nivel de Datos para el usuario
Pentium con DMA y discos SCSI	16MB+	10-20	Procesador de palabras; Firewall; Nivel de Datos para el usuario; Base de Datos.
Pentium con DMA y discos SCSI	16MB+	20+	Procesador de palabras; Firewall; Nivel de Datos para el usuario; Base de Datos; Firewall; Servidor de Impresión.
Pentium Pro con DMA y discos SCSI	32MB+	20-40+	Procesador de palabras; Firewall; Nivel de Datos para el usuario; Base de Datos; Firewall; Servidor de Impresión y Archivos para Windows 9x/NT.

Otro punto importante para correr el X es el soporte para tarjetas VGA. La mayoría de conjuntos de chips y tarjetas VGA son soportados con el ET3000, ET4000, GVGA, PVGA1, WD890c00, TRIDENT, CIRRUS, NCR y COMPAQ. La versión monocromática del X es llamada X386mono. Este servidor soporta tarjetas genéricas VGA y Hércules y además un ratón de bus o serial; de dos o tres botones.

Algunas distribuciones de Linux

Hay distribuciones de disponibilidad para escoger Linux, pero puede ser un poco difícil la elección. Las diferentes distribuciones reflejan las diferentes filosofías de cómo está trabajando este sistema operativo(1).

Slackware

Slackware tiene instalación y configuración directa generalmente y una buena aproximación lineal.

Los procesos de iniciación usan un simple menú del sistema y esto no puede fácilmente romperse. Se divide en discos, reflejando los días cuando la distribución comenzó con floppies, desde el cual escoges los componentes que se necesite instalar.

Desafortunadamente Slackware no provee ninguna comprensiva actualización mecánica. Si se necesita actualizar el sistema, básicamente reinstalará. De instancias, si se necesita actualizar un sistema como Samba, el sistema Slackware escribirá sobre los archivos comunes que están en el sistema, ya que se tendrá que volver a reconfigurar.

Caldera

Cuando Caldera inicia, insertaron Netscape a está distribución. Fue un serio intento a destinar aplicaciones de usuario final con un producto popular y que esto corriera bajo Linux, que era un consumo-orientado. Básicamente tiene 3 niveles el producto en Línea Open Linux: *Lite, Base y Standard.*

Debian

La distribución Debian es producida por una organización no lucrativa, software de interés público, corrientemente tiene algunos 200 desarrolladores voluntarios.

Si no se está usando Debian, este tiene una alta-calidad de reputación. Se usa un manejador de paquetes que tiene agradecimiento de los usuarios Linux y los desarrolladores consideran superior a RPM.

Yggdrasil

Yggdrasil es la vieja distribución basada en CD-ROM. Fue pionera de este método y esta muy estable en el kernel de Linux. Yggdrasil tiene un buen sistema de Ventanas X basado para instalación y configuración del sistema.

RedHat

RedHat está muy activo y tiene un valor-agregado para Linux en su instalación y administración. Esta distribución tiene una instalación completa y actualización basadas en el sistema de Manejador de Paquetes de Redhat (RPM). Redhat incluye un sistema gráfico de manejador de ventanas (ejemplo: Control-Panel). RedHat tiene un simple y directo sistema de instalación y actualización. RedHat tiene un buen nivel de profesionalismo y será rival para los sistemas que operan en las estaciones de trabajo.

Sitios donde encontrar más información acerca de Linux. Se dará algunas ligas y direcciones para distribución de Linux.

Distribuidor	Dirección URL
Caldera Inc.	http://www.caldera.com
Craftwork Solutions, Inc.	http://www.craftwork.com
Debian[1]	http://www.debian.com
DOSLINUX[1,2]	http://metalab.unc.edu/pub/Linux/distributions/dosLinux
InfoMagic, Inc.	http://www.infomagic.com
Linux System Labs (LSL)	http://www.lsl.com
Pacific HiTech	http://www.pht.com
Redhat Software, Inc.	http://www.redhat.com
S.uS.e.	http://www.suse.com
Trans-AmeriTech	http://www.zoom.com/tae
Walnut Creek CD-ROM[3]	http://www.cdrom.com
WorkGroup Solutions, Inc.	http://ftp.wgs.com/pub2/wgs
Yggdrasil Computing, Inc.	http://www.yggdrasil.com

[1]No comercial
[2]Corre bajo DOS
[3]Distribuidor Oficial de Slackware.

CAPITULO 1

1.1 INSTALACIÓN Y ACTUALIZACIÓN

En esta sección se dará la forma de realizar una instalación del sistema operativo Linux, debemos tener en cuenta las necesidades que se desean, el equipo con el que se cuenta y de esta forma realizar una instalación adecuada para tener un mayor desempeño(19).

Para esta instalación se necesita conocer la computadora y los periféricos que posee, por ejemplo:

1) Saber cuánta capacidad de memoria RAM se tiene en la computadora.

2) El tipo de mouse con que cuenta (serial ó de bus).

3) El tipo de monitor y los rangos en frecuencia horizontal y vertical.

4) Capacidad del disco duro.

5) Conocer la tarjeta de red.

6) La dirección IP de la computadora (Sí tiene acceso a Internet que no sea por modem).

7) Tarjeta de video (En caso de necesitar sesión gráfica, lo cual requiere un mínimo en espacio en disco).

La forma en que se realizará la instalación será a través de un floppy, este no puede tener sectores dañados, ya que si tiene alguno, el copiado aunque parezca tener éxito no lo tendrá, los siguientes archivos: *"boot.img" y "rescue.img"* son para realizar la instalación y el rescate del sistema respectivamente. Si estamos en una computadora Linux podemos teclear lo siguiente:

```
dd  if=boot.img of=/dev/fd0     <enter>
```

Otra opción es si tenemos el CDROM de instalación se puede hacer un booteo desde el mismo.

Comenzaremos con la instalación para la configuración del kernel.

Arranquemos el sistema con el disco creado (archivo boot.img ó bootnet.img) de Linux. En la pantalla aparecerá la siguiente información:

```
LILO boot:
```
En este momento debemos teclear tan solo <enter>

En la siguiente pantalla nos da la bienvenida y se nos dice dónde encontrar el manual de instalación. La siguiente pantalla nos pregunta si necesitamos soporte PCMCIA, lo cual en la PC no los contiene.

La próxima pantalla nos pregunta el método de instalación (CD-ROM, NFS image, Disco Duro, HTTP ó FTP, esto depende del disco de instalación que usemos, boot.img o bootnet.img), entonces le pondremos en modo **NFS** (Network File System).

En la próxima pantalla se especifica el módulo de opciones para la tarjeta de red y se tienen como opciones: Opciones especificas y autoprobado

En las opciones específicas nos pregunta la interrupción (IRQ), la entrada/salida de base (I/O) y misceláneas de opciones donde tenemos la información a la mano que es más lento, por eso usamos el autoprobado.

En la configuración de TCP/IP (Transmission Control Protocol / Internet Protocol), lo cual se pregunta:

configure TCP/IP	Escribiremos lo siguiente:
IP address: (Dirección IP)	*www.xxx.yyy.zzz IP de la computadora*
Netmask: (Máscara)	*(el valor lo coloca automáticamente, pero el usuario*
Network address:	*lo puede modificar).*
Broadcast address:	*Idem.*

En la otra ventana se presenta:	Ejemplo
El nombre del dominio	*undominio.nuestro*
El nombre del hosts	*nombre.undominio.nuestro*
Default gateway	*aaa.bbb.ccc.ddd*

En la siguiente opción se pondrá el tipo de medio por ejecutar :

En la siguiente ventana se pregunta si vamos a hacer instalación(Server, WorkStation, etc.) o actualización :
-En este caso se pide **Install** (instalación).

Nota: La opción Install destruye el contenido del *filesystem* raíz (/). Si se desea conservar lo anterior se selecciona la opción *UPDATE,* para agregar al sistema operativo.

Después dice si tenemos adaptadores SCSI:

La siguiente tabla indica la partición del lo(s) disco(s) y el dispositivo que debemos elegir: **/dev/hda, /dev/hdb, etc. ,** con la tecla **TAB** y después con la misma tecla se elige la tecla del monitor que dice **EDIT** y se teclea <enter>:

Ahí aparecerá un menú que en realidad es el comando **fdisk** que sirve para particionar el disco duro. Para la ayuda se hace uso del comando **man** y la letra **m** desplegará:

Comando	Acción
A	Activar la partición boteable
B	Edita la etiqueta del disco BSD
C	Activa la bandera de compatibilidad DOS
D	Borrar una partición
L	Lista los tipos de partición conocidos
M	Imprime es menú
N	Agrega una nueva partición
P	Imprime la tabla de partición
Q	Salir sin salvar los cambios
T	Cambiar una tabla de partición
U	Cambia el display
V	Verificar la tabla de partición
W	Escribir en la tabla de partición
X	Funciones extras (expertos solamente)

Con esto podemos crear particiones principalmente, cambiar el tipo de partición, borrar, ver los cambios, etc. En la instalación del sistema operativo LINUX se necesitan dos particiones mínimas (la del espacio en *swap* y la del sistema operativo). Para el área de *swap* tenemos que crear una partición del *doble* de la memoria RAM de que se dispone en la computadora. Para ver si hay particiones, con el comando print teclea la letra **p** y ahí aparecerán las diferentes particiones (si las hay) en lo que se tiene instalado, si no las hay solamente aparecerá (un ejemplo):

Disk /dev/hda: 32 heads, 63 sectors, 973 cylinders
*Units = cylinders of 2016 * 512 bytes*

Device Boot Begin Start End Blocks Id System

Command (m for help):

Pero si tenemos particiones aparecerá (como ejemplos):

Disk /dev/hda: 64 heads, 63 sectors, 528 cylinders
*Units = cylinders of 4032 * 512 bytes*

Device	*Boot*	*Begin*	*Start*	*End*	*Blocks*	*Id*	*System*
/dev/hda1	*	*1*	*1*	*264*	*532192+*	*6*	*DOS 16-bit >=32M*
/dev/hda2		*265*	*265*	*273*	*18144*	*82*	*Linux swap*
/dev/hda3		*274*	*274*	*528*	*514080*	*83*	*Linux native*

Command (m for help):

Otro ejemplo:

Disk /dev/hda: 32 heads, 63 sectors, 973 cylinders
*Units = cylinders of 2016 * 512 bytes*

Device	*Boot*	*Begin*	*Start*	*End*	*Blocks*	*Id*	*System*
/dev/hda1		*1*	*1*	*49*	*49360+*	*82*	*Linux swap*
/dev/hda2		*50*	*50*	*557*	*512064*	*83*	*Linux native*
/dev/hda3		*558*	*558*	*973*	*419328*	*83*	*Linux native*

Command (m for help):

Ahora para crear una partición se oprime la letra **n** <enter>:

Saldrá un menú :

Command action
e extended
p primary partition (1-4)

Como es disco se oprime la letra *p* (partición primaria) <enter>:

Aparecerá el número de partición (1 a 4), ya que solo acepta cuatro particiones "físicas"
Partition number (1-4):

Si no tenemos particiones teclearemos el número 1, de lo contrario se tecleará el número siguiente a la última partición.

First cylinder (1-977):

Aquí nos pide él número de cilindro correspondiente a la partición

Last cylinder or +size or +sizeM or +sizeK ([1]-977):

En la línea anterior se pide el número de cilindro final de la partición; o bien se puede especificar el tamaño de Kilobytes, en este caso que sea Kilo ó Mega y se pondrá la sintaxis +tamañoM ó +tamañoK, así seguirá hasta terminar de crear sus particiones, puede corroborar esto al ejecutar el comando print teclea la letra **p**. No olvidar el área *swap*.

Al terminar lo anterior es necesario cambiar el tipo de partición del área que usaremos como *swap*. Existen varios tipos (ejemplos: el área de *swap*, DOS, OS/2, etc.) Para conocer los tipos se ejecuta el comando *list*, tecleemos l <enter>:

Donde aparecerán sus números (código hexadecimal) y correspondientes tipos :

0 Empty	9 AIX bootable	75 PC/IX	b7 BSDI fs
1 DOS 12-bit FAT	a OS/2 Boot Manag	80 Old MINIX	b8 BSDI swap
2 XENIX root	40 Venix 80286	81 Linux/MINIX	c7 Syrinx
3 XENIX usr	51 Novell?	82 Linux swap	db CP/M
4 DOS 16-bit <32M	52 Microport	83 Linux native	e1 DOS access
5 Extended	63 GNU HURD	93 Amoeba	e3 DOS R/O
6 DOS 16-bit >=32	64 Novell Netware	94 Amoeba BBT	f2 DOS secondary
7 OS/2 HPFS	65 Novell Netware	a5 BSD/386	ff BBT
8 AIX			

Command (m for help):

para cambiar el tipo de partición ejecutamos el comando type (tecla *t*) :
que pedirá la partición que se quiere cambiar:

Partition number (1-4):

Y pide el código Hexadecimal (el cual se vio al ejecutar el comando List).

Por último necesitamos especificar cuál será la partición activa. La partición activa es la que busca el BIOS para cargar el sistema operativo, se activa con el comando Active, tecleando la letra **a**. Pedirá la partición donde se quiere que bootee y al darle <enter> no se mostrará en pantalla, pero si se ejecuta el comando print, tecleando la letra **p**, se verá donde está BOOT la partición que se selecciono para BOOT, tendrá un "*"(asterisco).

Disk /dev/hda: 64 heads, 63 sectors, 528 cylinders
*Units = cylinders of 4032 * 512 bytes*

Device	*Boot*	*Begin*	*Start*	*End*	*Blocks*	*Id*	*System*
/dev/hda1	*	*1*	*1*	*264*	*532192+*	*6*	*DOS 16-bit >=32M*
/dev/hda2		*265*	*265*	*273*	*18144*	*82*	*Linux swap*

/dev/hda3 274 274 528 514080 83 Linux native

Command (m for help):

NOTA: No se deben elegir dos particiones para booteo, ya que cuando arranque el sistema por primera vez no se sabrá cuál elegir y no responderá la computadora.

Para salvar los resultados se ejecuta el comando write, la letra w; si no se quiere salvar los cambios por algún motivo se ejecuta el comando quit, letra q.

Después de teclear w saldrá un menú en donde se quedo el disco de partición y con la tecla **TAB** se dirige al botón de **DONE** y teclea <enter>.

Aquí pide la activación del espacio de *swap* y elegirá automáticamente la se haya creado.

OJO: Si no se creó la partición de espacio de swap se tendrá que regresar al programa de partición de disco para hacerla ya que si no está hecha no podrá continuar, (lo cual originará siempre un reboteo en la computadora.

Active swap space: [] /dev/hda1

*[*] /dev/hda1 y después seleccionar **Ok***

Por consiguiente debemos seleccionar la tabla de partición a donde se dejará el sistema (según la partición que se haya elegido), después de elegir aparecerá otra ventana en la pantalla y se pide el punto de montaje en qué directorio se bajará y guardará la información, por *default* se pone en la partición que tenga en el *mount point* (punto de montaje) que es la diagonal "/".

[*] /dev/hda2

[*] check for bad blocks during format

Lo que se pide en los corchetes anteriores es la confirmación que de ahí se instalará el S.O. que posteriormente será el directorio raíz "/", en el siguiente corchete nos indica que checará los bloques malos del disco durante el formato y con el ejemplo de unas líneas arriba se activa con la barra espaciadora y **TAB**.

Después se mostrarán los paquetes que se desea instalar, un ejemplo son:

C development
Development libraries
C++ libraries
Print server
NFS server
Anonymous FTP/Gopher server

X Window system
Base
Base/kernel
Documentation (principalmente los man-pages)
Libraries
Networking
Shells (principalmente csh)
X development
Emacs
Emacs con X Windows

Estos son una muestra si se quiere trabajar en modo texto, en disco los más importantes son:

Applications/Networking
Base
Base/kernel
Documentation (principalmente los man-pages)
Libraries
Networking
Shell (principalmente csh)

Para las opciones anteriores, si aparecen con "*" (asterisco) se habilitan con la barra espaciadora, sino se deshabilitan con tal barra.

Para la instalación de las demás opciones, en el submenú de base se deberán elegir las opciones correspondientes para la instalación correcta.

Ahora si se escoge correctamente los paquetes, aparecerá una ventana que dirá que faltan paquetes y el programa los bajará automáticamente, esto está bien si tenemos espacio en disco, de lo contrario debemos rectificar, con la opción **CANCEL**, que nos regresará al menú de opciones para cargar los programas de instalación. De lo contrario aparecerá otra ventana donde dice que el sistema lo bajará a un archivo temporal y después lo grabará en disco, para esto tecleamos <enter>.

En el siguiente recuadro se estarán creando los sistema de archivos en el disco duro, lo que tardará algunos minutos y después bajará la información cuando aparezcan dos barras azules y poco a poco empezarán a llenar (indicando la información que se está grabando).

Después de que acaba de transmitir la información pregunta por el tipo de mouse (aparecerá un menú con los tipos de mouse), hay dos tipos principales; los de bus y los que se conectan por el puerto serial.

Si existe un mouse conocido por el S.O. Linux lo detectará. Posteriormente nos pregunta la sintaxis que si emulamos tres botones (Esto es muy conveniente para copiar, pegar y ejecutar, el tercer botón lo simula oprimiendo los dos botones al mismo tiempo).

Después configuraremos la tarjeta de video.

Nota: Las versiones de RedHat 5.0 y posteriores tienen plug and play que reconocen gran cantidad de tarjetas de video y aceleradores.

En la próxima ventana pregunta el tipo de monitor (la frecuencia horizontal y vertical), o si es más conveniente en este menú elegir la opción *custom*, si es que éste no lo detecta.

En la pantalla se pide los bits por pixel (bpp) con que se quiere trabajar en sesión gráfica (accionar todas las opciones que prefieras). En la siguiente ventana pregunta por el clockchip el cual puede ser detectado automáticamente, pero si sabe que clockchip tiene su computadora y se le encuentra en el menú, se selecciona, de lo contrario se teclea <enter> y lo auto detectará. En la siguiente ventana preguntará si desea cambiar la configuración de red (lo cual realizó anteriormente), si se hace correctamente teclea (Keep the configurator) y guardará la configuración.

Para reconfigurar la red de la computadora nos pedirá el número IP y todo lo correspondiente a la red que se vio páginas atrás.

Después pide el tipo de zona horaria, el cual es según de la región geográfica donde se encuentra el usuario, para que trabaje el huso horario local (México General).

En las siguientes ventanas pide la configuración del teclado (en qué idioma queremos trabajar, si es en español la opción será *es* y si inglés será *us*).

Después pide el passwd que necesita el superusuario (root), para su próxima entrada (No dar contraseñas obvias ni cortas).

La próxima ventana pide el cargador para instalar el sistema operativo (LILO BOOT), donde pide la partición en que se guardará dicho cargador. Aparecen varios casos:

/dev/hda	*Disco duro general*
/dev/hdax	Partición de la instalación en Linux
/dev/fd0	Floppy

1.2 INTRODUCCIÓN A REDES

1.2.1 Historia

La idea de red es probablemente tan vieja en cuanto a telecomunicaciones mismas. Considerando la gente viviendo en la edad de piedra, donde los tambores se pueden haber utilizado para transmitir mensajes entre individuos. Hoy en día tenemos las computadoras para dialogar y transmitir bajo fibra óptica y microondas, por mencionar algunos ejemplos.

Como en el año 1838 que inicia.la era de la comunicación de datos con el "telégrafo" por Samuel Morse. En 1864 se inician los trabajos de sincronización con la máquina de escribir, con el desarrollo de la teleimpresora, así sucesivamente se fueron descubriendo la manera de comunicar hasta nuestro días y en este caso no enfocamos a las redes de computadoras y sus protocolos de comunicaciones(10).

Será descrito el protocolo de comunicaciones TCP/IP. Este protocolo tienen la función de transmitir paquetes de datos para 2 computadoras. En cuanto a LINUX el modelo referencial OSI y el protocolo IPX es de manera fundamental.

Se define una red como la colección de "hosts" (anfitriones) que está comunicándose con cada red, pero al hablar de redes tenemos que hablar de INTERNET, así que haremos una breve historia.

1.2.2 La estructura de Internet

En realidad INTERNET es una red de redes de computadoras que intercambian información entre sí . De hecho, la palabra INTERNET se deriva del término internetwork, que significa "trabajo de redes". Una forma más fácil de visualizar INTERNET es imaginarla como una gran nube de información.

INTERNET surgió a partir de un programa de Investigación llamado Agencia de Proyecto de Investigación Avanzada para la Defensa de los Estados Unidos (DARPA, por sus siglas en inglés). A raíz de esta investigación surgió la primera versión de INTERNET, conocida como ARPANET. La ARPANET se inició en 1969. Este Departamento de la Defensa comisionó la construcción de la red de comunicaciones de computadoras que pudieran sobrevivir aun con la destrucción de unos o varios nodos. Como la historia lo marca, el DOD quería diseñar una red que pudiera sobrevivir a un ataque nuclear.

Para soportar la comunicación electrónica (e-mail), file sharing y acceso remoto a otras computadoras necesarias para los E.U., en función de un evento de ataque nuclear. El equipo de desarrollo de ARPANET creó el TCP/IP (Transmision Control Protocol / Internet Protocol). Sin embargo, la ARPANET no existió por más tiempo, pero legalmente vive en este protocolo.

1.2.3 La nfsnet

A finales de 1980, la NFSNET (National Science Foundation) anunció un proyecto para crear sitios de investigación de supercomputadoras en las universidades a lo largo y ancho de los E.U., como parte de este proyecto, NSF también comisionó una red de comunicaciones para unir todas estas supercomputadoras. Sin embargo, la vieja infraestructura de ARPANET fue inadecuada para esta tarea, pero en materia de protocolos, el TCP/IP, sería de gran utilidad.

Usando estos protocolos, la NSF desarrolló una red con estructura de tres tipos de nodos. El primer nodo fue un *"backbone"* que corría por todo el país. El segundo era un grupo de pequeñas redes regionales que conectaba universidades en una cierta región, que a su vez estaban conectadas al primer nodo del *"backbone"* . El tercer nodo consistía en conectar la LAN en una universidad, para permitir que los usuarios se pudieran conectar a las redes regionales y eventualmente al *backbone* NSF. La red fue un gran éxito y crecía exponencialmente. Había, sin embargo, un problema, se podía usar únicamente para propósitos de investigación académica. En otras palabras, las corporaciones que querían utilizar la NSFNET recibían un rotundo no.

1.2.4 El intercambio comercial internet (CIX)

Al reconocer el potencial comercial de algo tan funcional equivalente a la NSFNET, un par de compañías que construían y operaban redes regionales, llamadas UUNET, PSI y CERFNET, decidieron construir su propio *backbone* específicamente para su uso comercial. Para asegurar que los usuarios en UUNET pudieran hablar con usuarios en CERFNET, estas compañías se unieron para comisionar la construcción de un punto de encuentro llamado Comercial Internet Exchange (CIX), en el cual ellos pudieran intercambiar información. Esta red comercial proveía los servicios de la vieja red orientada a la investigación NSFNET. Estas redes crecían en nodos y en tamaño, especialmente por la popularidad del World Wide Web.

La comunicación es imposible sin algunas clases de lenguajes o código. En redes de computadoras, estos lenguajes son colectivamente referidos a protocolos.

1.2.5 Introducción del protocolo TCP/IP

Algunas características que han hecho tan popular el conjunto de protocolos de TCP/IP y que los distinguen de otros protocolos son:

- La independencia del hardware empleado en la red física, que permite trabajar sobre Ethernet, Token Ring, líneas seriales y mantener intactas las aplicaciones que corren en él, aun cuando la topología de la red sufra algún cambio.

- Los protocolos que conforman a la familia de TCP/IP, es decir, especificaciones técnicas disponibles públicamente sin costo alguno, aunque no las implementaciones disponibles y estandarizadas (desempeño consistente).

- Una de las características más importantes del TCP/IP es que permite que toda computadora en red pueda ser accesada directamente, ya que cuenta con una dirección única en toda la red mundial.

TCP/IP maneja el problema de transferencia de datos entre computadoras por medio de un modelo de referencia de 4 capas, cada una de las cuales describe los distintos procesos implicados:

Aplicación
Transporte (hosts)
Internet (network)
Data Link

- Capa de red o hardware

Sirve de interfaz entre la red LAN ó WAN y la capa de INTERNET de forma que oculta los detalles de conexión física a los protocolos de las capas superiores, asegurando así total independencia del hardware utilizado.

- Capa de Internet o network

Se encarga de la comunicación entre computadoras anfitrión basándose fundamentalmente en el protocolo IP. El servicio de conexión que brinda IP es conocido como no orientado a conexión, inconfiable (IP no asegura que el "datagrama" llegue a su destino) y de mejor esfuerzo (solo en casos extremos se descartará el paquete).

- Capa de Transporte o host

Brinda la comunicación entre programas de aplicaciones de punto a punto (end to end communication) con base en los protocolos UDP y TCP. El primero es una extensión del servicio no orientado a conexión e inconfiable que brinda IP. En cambio TCP ofrece un esquema confiable (los datos llegan sin error y TCP cuenta con mecanismos integrados para la detección y corrección de errores, si fuese necesario, para la retransmisión de paquetes), orientados a conexión (conection oriented, independiente del orden en que los datos hayan llegado a su destino, serán procesados con la misma secuencia con la que fueron transmitidos).

- Capa de Aplicaciones

Se ocupa de los servicios destinados al usuario tales como el correo electrónico, transferencia de archivos (FTP), sesión remota, etc.

1.2.6 Servicios soportados por TCP/IP

A continuación se listan los servicios soportados por TCP/IP:

- Conexión de terminal remota (Telnet).
- Transferencia de archivos (FTP)
- Correo electrónico (SMTP)
- Ejecución de procesos remotos (RPC)
- Administración y monitoreo de dispositivos (SNMP)
- Sistema gráfico distribuido (X.11)
- Sistema de nomenclatura de dominio (DNS)
- Sistema de archivo en red (NFS)

1.2.7 Direcciones IP

Como se ha mencionado por la red IP, el protocolo es un número de 32-bit. Cualquier computadora tiene asignada un número único en la red. Por dar un ejemplo, la dirección IP está dividida por 8-bit llamados bytes, si la dirección de cronos.fi-b.unam.mx tiene una dirección IP de 0x954xc0c04, está escrito como 132.248.59.2.(10)

Dependiendo del tamaño de la red, el anfitrión parte de que se pueden necesitar diferentes clases de redes en cuanto a su tamaño :

Clase A. La clase A comprende redes de 1.0.0.0 hasta 127.0.0.0. El número de redes está contenida en el primer byte. Está provista de 24-bit y contiene alrededor de 1.6 millones de anfitriones.

Clase B. La clase B contiene redes de 128.0.0.0 hasta 191.255.0.0. El número de red en el segundo 2 byte. Esta aloja de 16320 hasta 65024 anfitriones.

Clase C. La clase C tiene un rango de 192.0.0.0 hasta 233.255.255.0. Con el número de red contiene en el 3 byte. Esto aloja 2 millones de redes con 254 anfitriones.

Clase D, E y F. Estas direcciones abarcan el rango de 224.0.0.0 hasta 254.0.0.0. Son redes experimentales, que están reservadas para un futuro y no se deben especificar a alguna red.

La red 127.0.0.0 está reservada para el tráfico local del anfitrión. Usualmente, la dirección 127.0.0.1 será asignada a una interfase especial en el anfitrión.

1.2.8 Archivos de configuración TCP/IP

Existe un conjunto de archivos de configuración para la conexión en red TCP/IP en LINUX, de los que se hablará con detalle a continuación:

El archivo /etc/hosts. Toda computadora de una red TCP/IP tiene una dirección IP; un nombre de anfitrión canónico y cero o más un alias de nombres del anfitrión. El archivo /etc/hosts es el método original para transformar nombres de anfitrión en direcciones IP. Ahora daremos un ejemplo de una computadora:

```
#  /etc/hosts de la computadora kaos.fi-b.unam.mx
127.0.0 .1          localhost
# Esta computadora es
132.248.59.26    kaos.fi-b.unam.mx   kaos      #Computadora local
```

```
# Otras computadoras en la red.
132.248.59.1 aries.fi-b.unam.mx    aries      #El servidor de correo
132.248.59.2 cronos.fi-b.unam.mx  cronos      #El servidor de Archivos
132.248.59.6  odin.fi-b.unam.mx      odin      #El servidor de WEB
```

La dirección IP 127.0.0.1 se conoce como la dirección del ciclo de retorno local y está reservada para este objetivo. Es normal que se le asigne el nombre *localhost*. Si se piensa usar la computadora sólo como sistema aislado o se empleará SLIP ó PPP para conectarse con el exterior, se necesita nada más la dirección localhost en el archivo de anfitriones(4).

El archivo /etc/networs. Este archivo proporciona una lista de direcciones IP, nombres de redes en Internet. Cada línea proporciona la información de una red especifica, como sigue:

```
#Nombre de la Red Dirección IP
loopback          127.0.0.1
fi-b.unam.mx      132.248.59
fi-a.unam.mx      132.248.54
fi-p.unam.mx      132.248.52
Montreal.widgets.ca  198.73.139
```

El archivo /etc/protocols. El archivo /etc/protocols proporciona una lista de los protocolos Internet DARPA conocidos. Este archivo no debe modificarse, ya que la información es proporciona por el Centro de Información de Redes DDN. Como se muestra a continuación, cada línea contiene nombre y número de protocolo, así como cualquier *alias* para el protocolo.

```
# protocolos Internet (IP)
#
ip    0     IP    #Protocolo Internet, número de pseudoprotocolo
icmp  1     ICMP  #Protocolo de mensajes de control Internet
ggp   3     GGP   #Protocolo de Gateway a Gateway8
tcp   6     TCP   #Protocolo de control de Transmisión
egp   8     EGP   #Protocolo de gateway exterior
pup   12    PUP   #Protocolo universal de paquetes PARC
udp   17    UDP   #Protocolo de datagrama de usuario
hello 63    HELLO #Protocolo de enrutamiento HELLO
```

El archivo /etc/services. Proporciona una lista de los servicios disponibles en el host. Por cada servicio, en el archivo debe haber una línea que proporcione la siguiente información:

```
Nombre oficial del servicio
Número del puerto
```

```
Nombre del protocolo
Alias
```

Al igual que con los otros archivos, cada entrada se separa mediante espacio o tabulador. El número del puerto y el nombre del protocolo se consideran como un solo elemento, ya que se usa una diagonal (/) para separarlo. El siguiente ejemplo del archivo /etc/services :

```
#
#Servicio de red, estilo Internet
#
echo        7/tcp
echo        7/udp
discard     9/tcp       sink  null
discard     9/udp       sink  null
systat      11/tcp      users
ftp         21/tcp
telnet      23/tcp
smtp        25/tcp      mail
time        37/tcp      timeserver
time        37/udp      timeserver
rlp         39/udp      resource          #ubicación del recurso
whois       43/tcp      nicname
domain      53/tcp      nameserver  #servidor de nombres de dominio
domain      53/udp      nameserver
```

Es obvio que el funcionamiento de este archivo depende de la información en /etc/protocols. Si el servicio no está disponible, o si se quiere eliminar el soporte de un servicio específico, se puede comentar la línea apropiada con el signo de comentario. Sin embargo, en muchos casos el archivo /etc/inetd.conf también tiene que actualizarse para desactivar el soporte de un protocolo dado.

El archivo /etc/inetd.conf. Se usa para proporcionar la información al comando inetd. Como se verá más adelante el inetd es el superservidor de Internet. Atiende en puerto TCP/IP especificado, e inicia el comando apropiado cuando se solicita una conexión en dicho puerto. Esto ahorra recursos del sistema, ya que los *daemons* se arrancan sólo cuando se necesitan.

```
ftp     stream    tcp nowait root /usr/etc/in.ftpd     in.ftpd
telnet  stream    tcp nowait root /usr/etc/in.telnetd  in.telnetd
rsh     stream    tcp nowait root /usr/etc/in.rshd     in.rshd
login   stream    tcp nowait root /usr/etc/in.rlogind  in.rlogind
#tftp   dgram     udp wait   nobody /usr/etc/in.tftpd  in.tftpd
/tftpboot/diskless
```

Una entrada de este archivo consiste en una única línea compuesta de los siguiente campos:

servicio tipo protocolo espera usuario servidor línea_de_comando

Servicio: Proporciona el nombre del servicio. El nombre del servicio debe ser traducido a un número de puerto consultando en el fichero /etc/services.

Tipo: Especifica un tipo de socket, ya sea *stream* (para los protocolos orientados a la conexión) o *dgram* (para protocolos no orientados a la conexión). Servicios basados en TCP deberán, por lo tanto, usar siempre *stream*, mientras que los basados en UDP requieren *dgram*.

Protocolo: Indica el protocolo de transporte usado por el servicio. Este debe ser un nombre de protocolo válido que se pueda encontrar en el fichero protocolos, también descrito más adelante.

Espera: Esta opción se aplica sólo en sockets de tipo *dgram*. Puede tomar los valores de *wait* y *nowait*. Si se especifica el *wait*, *inetd* ejecutará solo un servidor cada vez para el puerto especificado. De otro modo se continuará escuchando por el puerto inmediatamente después de ejecutar el servidor.

Esto es útil para servidores "single-threaded" que leen todos los datagramas que entran hasta que no llegan más, y después acaban. La mayor parte de los servidores RPC son de este tipo y deberán por ello especificar *wait*. El otro tipo de servidores, los "multi-threaded", permiten un número ilimitado de instancias corriendo concurrentemente; éstos son raramente utilizados. Estos servidores deberán especificarse como *nowait*.

Con sockets de tipo *stream* se deberá especificar siempre *nowait*.

Usuario: Este es el identificador de login del usuario en el cual que se ejecutará el proceso. Por lo general, este suele ser el usuario **root**, aunque algunos servicios pueden usar diferentes cuentas. Es una buena idea el aplicar aquí el principio del menor privilegio, que indica que uno no debería ejecutar un comando bajo una cuenta privilegiada si el programa no lo requiere para funcionar correctamente. Por ejemplo, el servidor de noticias NNTP se ejecutará como *news*, mientras que otros servicios que podrían significar un riesgo para la seguridad (como *tfip* y *finger*) son normalmente ejecutados como **nobody**.

Servidor: Proporciona el camino completo del programa servidor a ejecutar. Los servicios internos son marcados como la palabra *internal.*

Línea_de_comando: Esta es la línea de comando ejecutada por el servidor. Esto incluye el argumento 0, es decir, el nombre del comando. Normalmente, éste será el nombre del programa del servidor, salvo que el programa se comporte de forma distinta cuando se le invoque con un nombre diferente.

Este campo se deja vacío para los servicios internos.

Como vemos renglones atrás, la línea de *tftp* está deshabilitada. *Tftp* establece el Trivial File Transfer Protocol que permite transferir cualquier archivo del sistema que tenga permiso de lectura global sin revisión de passwords, etc. Esto es peligroso con el archivo /etc/passwd, sobre todo si no se usa el shadow password.

Tftp es usado comúnmente por clientes y terminales X, sin unidad de discos para obtener su código de un servidor de arranque. Si por está razón necesita ejecutar *tftp*, asegúrese de limitar su acción a los directorios de donde los clientes obtendrán los ficheros, añadiendo esos nombres de directorio a la línea del comando del *tftpd.* Esto se muestra en la segunda línea *tftp* del ejemplo.

1.2.9 Manejo de los comandos *r*

Hay varios comandos para ejecutar comandos en modos remotos. Son rlogin, rsh, rcp y rcmd. Todos ellos lanzan un shell en el nodo remoto y permiten al usuario ejecutar comandos. Por supuesto, el cliente necesita tener una cuenta en el nodo en el que se van a ejecutar los comandos. Por ello, todos estos comandos llevan a cabo un procedimiento de autorización. Normalmente, el cliente indicará el nombre de login del usuario al servidor, el cual requerirá un password que será validado de la forma habitual(16).

A veces, sin embargo, es deseable el relajar estas revisiones de autorización para ciertos usuarios. Por ejemplo, si se tiene que entrar frecuentemente en otras computadoras de la LAN, tal vez se desee ser admitido sin tener que escribir el password cada vez.

Deshabilitar autorizaciones sólo es aconsejable en un número reducido de nodos, cuyas bases de datos de passwords estén sincronizadas, o para un número reducido de usuarios privilegiados que necesiten acceder a muchas computadoras por razones administrativas. Siempre que se desee permitir a gente entrar en su nodo sin tener que especificar un login o password, debe asegurarse de que no se permite acceso accidentalmente a nadie más.

RLOGIN. Este comando conecta la sesión local a una sesión remota en un host. Para iniciar una sesión remota desde una terminal remota se usa las siguiente expresiones :

```
$rlogin computadora_remota
$rlogin -l usuario computadora_remota
```

El tipo de terminal de la conexión remota es el mismo que el empleado para las conexiones actuales, a menos que se modifique a través de los archivos de inicio del shell del usuario. Todo el eco de caracteres se hace en el sitio remoto así que, salvo las demoras, el uso de rlogin es transparente para el usuario. La terminación de la conexión se hace ya sea desconectándose del host remoto o mediante los caracteres de terminación, que son la tilde y el punto (~.).

RCP. El comando rcp, copia remota, permite al usuario copiar un archivo de un host a otro. Este comando copia archivos entre dos computadoras. Cada argumento de archivo o de directorio es, ya sea el nombre de archivo remoto de la forma "rhost:ruta" o bien el nombre de archivo local.

La sintaxis es la siguiente :

```
rcp [ -p ] file 1 file 2
rcp [ -p ] [ -r ] file .... directory
```

Ejemplos :

```
$rcp -rp /users          brahm:/users
$rcp test.txt    brahm:test.txt
```

RSH, REMSH y RCMD. Estos tres comandos tienen una función similar, que es la de ejecutar comandos en un sistema remoto. Los comandos interactivos no son buenos candidatos para este tipo de ejecución.

No debe confundirse la utilización rsh de ejecución remota con el shell restringido (rsh) que se encuentra en los sistemas System V Unix. Del mismo modo, algunas versiones del System V usan remsh en lugar de rsh. Por lo general, los sistemas que usan rsh para ejecución remota son sistemas Unix basados en BSD. El comando rsh funciona conectándose con el nombre de host especificado y ejecutando el comando especificado, rsh copia su entrada estándar al comando remoto, la salida estándar del comando remoto a su propia salida y el error estándar del comando remoto a su propio error. Las señales de

interrupción, salida y terminación se propagan al comando remoto; rsh normalmente termina cuando lo hace el comando remoto.

La sintaxis del comando es las siguiente:

```
$rsh -l username   -n   hostname   comando
$rsh hostname   -l username   -n   commando
```

La opción *-l username* se refiere al nombre de usuario como nombre de usuario remoto, en lugar del nombre del usuario local. A falta de esta opción, el nombre de usuario remoto es el mismo que al del usuario local.

La opción *-n r*edirecciona la entrada de rsh a /dev/null. En ocasiones necesitará esta opción para evitar interacciones desafortunadas entre rsh y el shell que lo llama. Por ejemplo, si se está ejecutando rsh e inicia un rsh en segundo plano sin redireccionar su entrada lejos de la terminal, se bloqueará aun cuando el comando remoto no envíe lecturas. La opción -n envía esto.

En el sistema remoto puede ejecutarse prácticamente cualquier comando. Sin embargo, los comandos que dependen de las características de la terminal o un nivel de la interacción del usuario no son buenos candidatos para usarse con rsh.

El comando rcmd es prácticamente idéntico a rsh, excepto que por lo general se encuentra en los sistemas System V. De hecho, rcmd tiene las mismas opciones y funciona del mismo modo que rsh en las BSD de Unix. El siguiente ejemplo ilustra un rcmd que accesa un sistema remoto al no especificar un comando cuando rcmd inicia.

```
$rcmd thor
$rcmd thor ps -ax | more
```

CAPÍTULO 2

2.1 SISTEMA DE ARRANQUE Y PARO

Cuando se realiza el arranque con el disco duro, normalmente sé esta llamando a un administrador en especial llamado LILO (Linux LOader). Lilo es un administrador de arranque de propósito general. Al hacerlo de esta forma, el kernel de Linux es el encargado de verificar todo el sistema(3).

Existe un archivo y un programa que el kernel ejecuta al momento de inicializar los dispositivos. Uno es el archivo `/sbin/init` y el otro es el `/etc/inittab`. El programa `/sbin/init` realiza nuevos procesos y restablece ciertos programas al momento de salir. Todo lo que `/sbin/init` realiza está controlado por el archivo `/etc/inittab`. En el archivo `/etc/inittab` se encuentran todos los parámetros de configuración de su sistema en particular los que se tienen instalados.

Al igual que el arranque en el momento de querer detener el sistema se puede realizar de varias formas: la primera es apagar el sistema (aunque no es recomendable ya que puede causar daños a los archivos), la segunda y más recomendada es utilizar el comando que aquí veremos.

El comando que utilizaremos para la detención del sistema es: `/sbin/shutdown`, su sintaxis es la siguiente:

`/sbin/shutdown` [indicadores] *tiempo* [mensaje]

donde:
mensaje es lo que se envía a todos los usuarios que están en sesión en ese momento *tiempo* es el momento en que va a suceder el paro.

Se tiene como indicadores las siguientes opciones:

Bandera	Descripción
-t *seg.*	Espera segundos para detener el proceso.
-k	En realidad no detiene el sistema sólo envía el mensaje.
-r	Vuelve a arrancar después del paro.
-h	Una vez que se realizó el paro se detiene.
-c	Cancela una detección que se encuentra en proceso.

Existen otros dos comandos en el sistema de Linux para realizar las tareas antes descritas, estas son:

`halt y reboot`

En el archivo '`/var/log/wtmp`' se guardan los cambios que se realizan con dichos comandos y ambos le dicen al kernel si debe parar o rebootear el sistema.

Cuando las instrucciones '`halt`' o '`reboot`' son llamadas y el sistema **no** se encuentra en el nivel de corrida 0 ó 6, el comando '`shutdown`' se ejecuta en su lugar (con la bandera -h o -r).

La estructura de los comandos es la siguiente:

```
/sbin/halt          [-n][-w][-d][-f]
/sbin/reboot        [-n][-w][-d][-f]
```

La descripción de las opciones se hace a continuación:

-n	No existe sincronía antes del '`halt`' o del '`reboot`'
-w	No hay actualización, pero se escribe un registro en el archivo '`/var/log/wtmp`'
-d	No se escribe al archivo wtmp. Esta bandera implica -n
-f	Fuerza el '`halt`' o el '`reboot`', no se hace una llamada a '`shutdown`'

NOTA: Cuando se especifica la bandera **-f** se hace la llamada con la señal 9, mientras que en el otro caso se realiza la llamada con la señal 15 y en este caso también se llama a las funciones '`startup`' y '`shutdown`' respectivamente.

Encienda el sistema e indique en el prompt la opción de cargar Linux. En su pantalla aparecerá la siguiente información:

```
LILO boot:
```

En este momento debe teclear tan solo <TAB> y ver las etiquetas que tiene la computadora:

```
LILO boot:
dos   Linux                 (Queremos iniciar con Linux)

LILO boot: Linux        (y posteriormente teclea <enter>)
```

Ahora se puede ir revisando lo que pasa en el sistema y que va apareciendo en el monitor.

Una vez que se han verificado todos los componentes del sistema, éste pasará del estado monousuario al estado multiusuario. En este momento el sistema está listo para aceptar sesiones de trabajo y se podrá entrar a sesión como super-usuario (root).

Ahora se puede utilizar el sistema para ejecutar los comandos que se necesitan.

Se comenzará con el apagado del sistema. Para esto hay que recordar que no es conveniente tan sólo apagar el sistema, ya que causaría problemas posteriores.

El comando `/sbin/shutdown` permite detener el sistema de forma conveniente, se teclea el siguiente comando:

```
# /sbin/shutdown -r now
```

Una vez que se reinicie, el sistema podrá cambiar los parámetros del comando anterior y comprobar las diferentes formas de apagarlo.

Ahora se verán los diferentes niveles que corre Linux y estos tienen una pequeña descripción en el archivo `/etc/inittab:`

```
0) Paro total del sistema.
1) Modo monousuario ó administrativo. En este estado se trabaja para la
revisión de los dispositivos.
2) Multiusuario sin NFS, donde no trabajan todos los programas en red.
3) Multiusuario completo.
4) No usado.
5) X11. El sistema X window se corre a este nivel.
6) Reboot. Donde primero se realiza el logout a los usuarios y después se
da de baja el sistema.
```

Cuando inicia el sistema por default la línea que se encuentra en el archivo `/etc/inittab`, indica en que nivel corre Linux:

```
id:3:initdefault:
```

donde:
id -> Identificador de la función a ejecutar.
3 -> Número de nivel en donde se ejecuta.
Initdefault -> Lo que representa (en esta cadena tienen mas significado, ver el páginas del manual para ver detalles).

Cada campo está separado por ":"

Ya que se inicia el sistema en modo multiusuario, se inicia con el usuario *root*.

Si se está en el sistema y se quiere darle mantenimiento (Nivel 1) entonces se teclea:

```
#/sbin/init 1                          (se verá que pasa)

bash#                                (indica el nivel monousuario)
```

Para regresar al nivel original, se usa el siguiente comando:

```
bash#/sbin/init 3
```

Y se regresa al nivel original.

Se reinicia el sistema (ya se tiene más de una opción). Cuando se tenga en pantalla el cargador de Linux, se teclea directamente el nivel: *init [no.]*

```
LILO boot: Linux init 1
...
...
Carga los driver´s, etc.
...
...
bash#                                  (indica el nivel 1)
```

En este nivel se harán todas las tareas administrativas, para salir y regresar al nivel por default solamente se dará el comando `exit`:

```
bash#exit     (regresamos al nivel 3 o al puesto por default).
```

Como se ha notado, se puede cambiar arbitrariamente a los diferentes niveles que tiene Linux. Cuando se hacen estos cambios de niveles, se da de baja unos servicios y se levantan otros.

Estos archivos son llamados ***scripts*** de configuración y para los sistemas Linux funcionan para dar de alta y baja servicios. Estos ***scripts*** son similares en función al archivo *AUTOEXEC.BAT* de *DOS*, pero estos son mucho más flexibles y potentes.

Redhat, como otras distribuciones, un caso es Debian, guarda tradición con el Unix System V en cuanto a los directorios `/etc/rc.d`.

El directorio *init.d* contiene casi todos los scripts de una configuración básica del sistema. Se hablará posteriormente de los ***scripts*** que están en este directorio.

El ***scripts*** *rc* (run control) es responsable de poner en función variables básicas como el *Hostname.* Aquí también está para iniciar servicios, cuando hacemos cambios de nivel.

El ***script*** *rc.local,* determina lo que aparece en el archivo `/etc/issue` entre otras cosas tales como variables o donde se puede agregar las variables o funciones.

El ***script*** *rc.sysinit* corre únicamente cuando el sistema está iniciando; inicia los ***scripts*** generales de red y activa la partición swap (intercambio).

Los directorios *rc0.d* al *rc6.d* contienen ligas que especifican ***scripts*** que se ejecutan cuando hay cambios de cada nivel.

Ahora se dará una breve descripción de los servicios que se encuentran dentro del directorio *init.d(5)*

atd. Inicia el *daemon at*, el cual ejecuta en un tiempo, comandos ó ***scripts***.
crond. Inicia el *daemon crontab*, el cual ejecuta tareas periódicamente.
functions. Contiene funciones usadas por otros ***scripts***.
gpm. Inicia el programa gpm, habilitando el ratón desde una pantalla de texto.
halt. Función para detener la computadora de reboot ó shutdown.
inet. Inicia la red TCP/IP. Configura la interfaz ethernet, pone el ruteo funcionando y otros servicios.
kerneld. Carga automáticamente los módulos del kernel cuando se necesiten.
keytable. Mapea el teclado.
killall.Para los *daemons* innecesarios.
lpd. Inicia y detiene el *daemon* de impresión lpd.
network. Para e inicia la red.
nfsfs. Monta y desmonta los sistemas de archivos remotos (NFS).
pcmcia. Inicia los adaptadores PCMCIA de las laptops.
random. Inicia el número generado aleatoriamente.
routed. Inicia el *daemon routed*, esto usa el protocolo RIP, automáticamente actualiza la tabla de ruteo.
rusersd. El *daemon rusers* ayuda a localizar usuarios en computadoras remotas.
rwhod. El *daemon rwho* lista a los usuarios loggeados en una computadora remota.
sendmail. Inicia y para el *daemon* **sendmail**, el cual transfiere mensajes e-mail a su respectivo destino.
smb. Inicia y para los servicios *samba*.
syslog. Inicia el sistema de *logging*. Esto es muy bueno para la seguridad y otras funciones administrativas, ya que todos estos recursos tienen auditoría.

2.2 ADMINISTRACIÓN DE USUARIOS

El administrador del sistema está encargado de la administración de los usuarios; ya que involucra el registro de los mismos, el ajuste de los privilegios, la creación y asignación de directorios personales, la asignación de usuarios a grupos, asignación de cuota y la eliminación de usuarios en caso necesario.

2.2.1 Alta de una cuenta de usuario

Las tareas que se llevan a cabo para generar una cuenta de usuario son:

I. Editar el /etc/passwd.
II. Crear un directorio de casa (HOME DIRECTORY).
III. Editar el /etc/group.
IV. Copiar archivos de configuración.
V. Asignar una contrascña (password).
VI. Asignar recursos.

En Linux, varias de estas tareas se llevan a cabo mediante el comando `'/usr/sbin/adduser'`

Edición del `/etc/passwd`

El archivo /etc/passwd es el archivo de registro de los usuarios válidos en el sistema. La información que contiene es la siguiente:

Nombre de la cuenta de usuario.
Password encriptado.
Identificador de usuario (UID).
Identificador del grupo de default (GID).
Información particular del usuario (GECOS).
Directorio de casa (HOME DIRECTORY).
Shell de inicio.

Ejemplos de registros del `/etc/passwd`:

```
sergio:HP37g3QhPYvyk:353:100:dominguez herrera sergio,,,:/users/ptc/sergio:/bin/ksh
gus:.PCVB95IGB8YM:355:100:El Gran Gus,,,:/users/ptc/gus:/bin/ksh
firebird:YFv0O04k5H7S2:358:100:padilla m. angel,,,:/users/ptc/firebird:/bin/ksh
gork:GAiWHeAqoKgRk:396:100:diaz jimenez jose,,,:/users/ptc/gork:/bin/ksh
rody:iY2PPSOlNN81g:463:100:mendez zarate rodolfo,,,:/users/ptc/rody:/bin/ksh
```

Si se quieres registrar al usuario Frida Kalo con el comando '/usr/sbin/adduser'. Para esto, se entra a sesión como super usuario y se teclea lo siguiente:

```
#/usr/sbin/adduser fridaka <enter>
Looking for first available UID... 507
Looking for first available GID... 501

Adding login: fridaka...done.
Creating home directory: /home/fridaka...done.
Creating mailbox: /var/spool/mail/fridaka...done.

Don't forget to set the password.
#
```

Se edita el archivo /etc/passwd y se localiza al nuevo usuario Frida Kalo.

```
#vi /etc/passwd <enter>
```

NOTA: Se puede usar el comando /usr/sbin/vipw

Dentro del editor, se escribe 'Frida Kalo' en el campo de información del usuario (GECOS) y salva el archivo. El registro debe quedar así:

```
fridaka:*:507:501:Frida Kalo:/home/fridaka:/bin/bash
```

NOTA: El UID y el GID pueden ser distintos.

Se asigna la contraseña a Frida Kalo utilizando el comando '/usr/bin/passwd'.

Se asigna la palabra 'redes00'. Para esto se teclea lo siguiente:

```
# passwd fridaka <enter>
New password:  Teclea redes99 <enter>
New password (again):     Repítelo <enter>
Password changed
passwd: all authentication tokens updated successfully
#
```

Para verificar que el password se ha asignado correctamente, se entra a sesión como el usuario Frida Kalo.

Se utiliza la secuencia de teclas '<ctrl> + <alt> + <F2>'.

Donde <F2> es la segunda sesión, <F3> la tercera y así sucesivamente. Hay que recordar que el *login* de Frida Kalo es fridaka y que su *passwd* es redes00.

Una vez que se logre entrar a sesión, debe observarse el contenido del registro de Frida Kalo, utilizando el comando 'grep':

grep "^fridaka:" /etc/passwd <enter>
Debe saltar a la sesión de fridaka.

2.2.2 Alta de una cuenta de grupo

En cualquier plataforma UNIX, se pueden crear grupos de trabajo, los cuales compartirán los mismos recursos. El registro se lleva a través del archivo '/etc/group'

Edición del /etc/group

Cada registro del archivo '/etc/group' contiene:

- Nombre del grupo
- Password (este campo ya no es utilizado)
- Identificador del grupo (GID)
- Lista de claves de usuario pertenecientes al grupo (separadas por comas).

El nombre de cada grupo está formado por 8 caracteres máximo.

No es necesario que las claves de usuario se indiquen como pertenecientes a su grupo de default en el '/etc/group'. Solamente se indica cuando una cuenta pertenece a grupos adicionales.

Ejemplos de registros del '/etc/group':

```
ptc::100:antonio,david,doom,falcon,firebird,gork,gus,invent,jess,mena
precand::101:prebe01,prebe02,prebe03,prebe04,prebe05,prebe06,prebe07
cursos::102:alfa01,alfa02,alfa03,alfa04,alfa05,alfa06,alfa07,alfa08
alum::103:gadc,morrol,alexcho,isaacm,tonio
profes::104:adrian,aedlot,ajim,ante,avel50,cgj001,coatza,donal,eco,garf70
die::105:anapat,apollo,cgarci,dial23,dms208,eamch,gabo,karo
informix::106:informix
sybase:*:200:sybase
veraltas::300:sergio,morfeo,kyo,patty,mena,greedo,rody,chesst,gus,falcon
```

Al cambiarse a la sesión de *root* (se presiona ctrl+alt+F1, o la F? adecuada) y se crea un grupo llamado 'amigos', que se asigna al usuario Frida Kalo (fridaka):

Primero, se edita el archivo de grupo:

```
#vi /etc/group
```

Segundo, se localiza el renglón (registro) que contiene a `fridaka`, el cual puede tener el siguiente formato: `fridaka::501:fridaka`

Tercero, se cambia el nombre del primer campo, por el nuevo grupo `'amigos'`: `amigos::501:fridaka`

Cuarto, se salva el archivo.

Se cambia a la sesión del usuario fridaka, se lista los archivos con el comando `'ls'`

Se crea un nuevo grupo, llamado `'equipo'`. La entrada en el archivo `'/etc/group'` debe quedar así:

```
equipo::777:
```

Como se puede ver, en el punto anterior se realiza un cambio de grupo con el usuario fridaka, pero no se cambio de grupo sus archivos, por lo que pertenecen al viejo grupo. Ahora, se cambiarán dichos archivos al nuevo grupo con el siguiente comando:

```
#find /home/fridaka -exec chgrp amigos {} \;
```

Nuevamente, se teclea el comando `'ls -lia'` y se observa a que grupo pertenecen. También se teclea el comando `'id'` y se verifica que 'Frida Kalo' si aparezca en el grupo `'amigos'`.

2.2.3 Cambios de UID, GID,login, shell y de directorio de casa

a) Cambio de UID

En muchas ocasiones y por distintas razones, es necesario cambiar a un usuario el UID que ya tiene asignado por algún otro. Esta tarea se lleva a cabo de la siguiente forma:

Primero, se encuentra un UID libre, es decir un UID que no esté asignado a algún usuario.

Segundo, se edita el archivo `'/etc/passwd'`, se localiza el registro del usuario `fridaka` y en su campo de UID (tercer campo) se cambia por 2125, se salva el archivo. El registro debe quedar así:

```
fridaka:rF/4sxUyEUVY:2125:501:Frida kalo:/home/fridaka:/bin/bash
```

Se entra a sesión como usuario `fridaka` y se verifica que el usuario pertenezca al UID se le asignó, es decir, al UID 2125, también se verifica que los archivos sean los correspondientes.

El primer paso para cambiar el UID de un usuario ya está hecho, pero falta que los archivos del UID anterior se cambien al UID nuevo, hay que recordar que esta información está almacenada en la estructura i-nodo. Para cambiar los archivos al nuevo UID se teclea el siguiente comando:

```
#find /home/fridaka -exec chown 2125 {} \;
```

De la sesión que se tenía como `fridaka` se salta y se vuelve a entrar. Se ejecutan los comandos `'id'` y `'ls -lia'` y se comparan los resultados con los obtenidos en el inciso 3.3.

b) Cambio de GUID

En muchas ocasiones y por distintas razones, es necesario cambiar a un usuario el GUID que ya tiene asignado por algún otro. Esta tarea se lleva a cabo así:

Primero, se define a qué nuevo grupo va a pertenecer y se obtiene el valor numérico. En nuestro caso ese nuevo grupo es el de `'equipo'` y cuyo valor es 777.

Segundo, se edita el archivo `'/etc/passwd'`, se localiza el registro del usuario `fridaka` y en el campo de GUID (cuarto campo) se cambia por 777, se salva el archivo. El registro debe quedar:

```
fridaka:rF/4sxUyEUVY:2125:777:Frida Kalo:/home/fridaka:/bin/bash
```

Los archivos del usuario `fridaka` pertenecen al grupo `amigos`. Se comprueba entrando nuevamente a sesión como `fridaka` y ejecuta los comandos `'id'` y `'ls -lia'`. Para cambiar de grupo a los archivos del usuario `fridaka` se ejecuta el siguiente comando:

```
#find /home/fridaka -exec chgrp 777 {} \;
```

Se entra nuevamente a sesión como usuario `fridaka`, si ya se estaba dentro; salta y se vuelve a entrar, y se ejecutan los comandos `'id'` y `'ls -lia'`. Se comparan los resultados con los obtenidos en el inciso 3.8.

c) Cambio de login

Se edita el archivo `'/etc/passwd'`, se localiza el registro del usuario `fridaka` y se cambia el login (primer campo) **'fridaka'** por **'apollo'**.

```
fridaka:rF/4sxUyEUVY:2125:777:FridaKalo:/home/fridaka:/bin/bash
```

sustituyendo tendremos :

```
apollo:rF/4sxUyEUVY:2125:777:Frida Kalo:/home/fridaka:/bin/bash
```

y se guarda el archivo.

Se abre una sesión como usuario Frida Kalo, utilizando el nuevo login. Ejecutando el comando 'id' se identifican y se anota los valores ID de usuario y grupo obtenidos.

d) Cambio de Shell

Una de las tareas más comunes del administrador es cambiar o adecuar el Shell de inicio a sus usuarios. Se edita el archivo '/etc/passwd' y modifica el séptimo campo del registro del usuario 'apollo' de '/bin/bash' por '/bin/sh'

Realizando los cambios se tiene:

```
apollo:rF/4sxUyEUVY:2125:777:Frida Kalo:/home/fridaka:/bin/bash
```

y :

```
apollo:rF/4sxUyEUVY:2125:777:Frida Kalo:/home/fridaka:/bin/sh
```

Estando en la sesión del usuario 'apollo', se ejecuta el comando '/usr/bin/chsh' y se cambia el Shell de '/bin/sh' a '/bin/bash'.

e) Cambio de directorio de casa

Otra de las actividades más comunes del administrador es cambiar el "directorio de casa" de los usuarios, ya sea porque se cambiaron el Login, de grupo, etc. Para realizar esta labor, primero se edita el archivo '/etc/passwd' . El nuevo directorio de casa será '/users/amigos/apollo',

```
apollo:rF/4AsxUyEUVY:2125:777:Frida Kalo:/home/fridaka:/bin/bash
```

deberá cambiarse por:

```
apollo:rF/4AsxUyEUVY:2125:777:Frida Kalo:/users/amigos/apollo:/bin/bash
```

Se crea el nuevo directorio '/users/amigos/apollo' y se cambia al directorio de casa del usuario apollo:

```
#mkdir -p /users/amigos/apollo
#cd /home/fridaka
```

Se copian los archivos del usuario al nuevo directorio:

```
#find . -print | cpio -pdmv /users/amigos/apollo
```

Se eliminan los archivos del directorio /home/fridaka

```
#cd ..
#rm -rf fridaka
```

Finalmente cambia de dueño y grupo al nuevo directorio '/users/amigos/apollo':

```
#cd ~apollo
#chown apollo .
#chgrp 777 .
```

2.2.4 Eliminación o baja definitiva de una cuenta de usuario

Otra de la actividades más comunes de un administrador es la de dar de baja a un usuario. Para realizar esta tarea se siguen los siguientes pasos.

Se crea un usuario temporal que tenga login **'dummy'** y se le asigna password. No importa el grupo al que pertenezca.

Desde una sesión como **'root'** se le envía una carta (e-mail) de bienvenida al usuario **'dummy'**.

Se abre una sesión como el usuario **'dummy'**, y se ejecuta el siguiente comando.

Se revisa el archivo de correo del usuario **'dummy'**, se ejecuta el siguiente comando:

```
#ls -lia /usr/spool/mail/dummy
```

Se cierra la sesión del usuario **'dummy'**, y se entra a sesión como **'root'**. Para borrar del sistema los archivos del usuario **'dummy'** utilizando el siguiente comando:

```
# find / -user dummy -print -exec rm -f {} \;
```

Finalmente, se edita el '/etc/passwd' y se borra la línea (registro) del usuario **'dummy'**. También se edita el '/etc/group' y se borra el UID del usuario **'dummy'** de la lista del grupo al que pertenece.

Se anotan los comandos y/o pasos que se siguieron para comprobar que el usuario **'dummy'** ha sido dado de baja definitivamente.

2.2.5 Configuración de cuota

Si se tiene a los usuarios en el sistema, y no se quiere que ocupen todo el espacio en disco, el administrador decidirá cuánto espacio en disco les proporcionará a cada usuario o a un grupo, para ello hay un programa que facilita esta tarea llamado ***quota***, el cual especifica la cantidad de espacio en disco por utilizar y los *inodos* que se pueden crear, esto es bastante bueno, ya que los usuarios pueden:

I) Cometer errores al compilar y crean un archivo de error (denominados core)
II)Archivos que bajan de Internet, etc.

Antes que nada los usuarios se deben crear en otra partición diferente al del sistema (" / "), con ello se logran dos cosas:

I) Si le pasa algo al sistema ó simplemente se quiere actualizarlo, los archivos de usuarios no se afectan.
II) Activar el programa **quota.**

Saber dónde están ubicados los usuarios. Un ejemplo es cuando los usuarios se encuentran en el directorio ***/users*** ó en la partición ***/dev/hda3*** se pueden ver con el siguiente comando:

```
#df
Filesystem 1024-blocks Used   Available Capacity  Mounted on
/dev/hda2   495746     318861 151282    68%       /
/dev/hda3   405963     33821  351176    9%        /users
/dev/hda4   949529     170803 729672    19%       /export
```

Después de localizar en qué directorio(s) están los usuarios, en el archivo */etc/fstab* se tiene que agregar una palabra:

```
/dev/hda2        /              ext2       defaults          1 1
/dev/hda1        swap           swap       defaults          0 0
/dev/fd0         /mnt/floopy    ext2       noauto            0 0
/dev/hda3        /users         ext2       defaults          1 1
none             /proc          proc       defaults          0 0
```

Agregando la palabra en la partición donde están los usuarios:

```
/dev/hda2        /              ext2  defaults          1 1
/dev/hda1        swap           swap  defaults          0 0
/dev/fd0         /mnt/floopy    ext2  noauto            0 0
/dev/hda3        /users         ext2  defaults,usrquota 1 1
none             /proc          proc  defaults          0 0
```

donde la palabra ***usrquota*** es para darle a los usuarios cuota, si se quiere dar a un grupo cuota, se pondrá ***grpquota***, si se quiere ambos se pondrá:

```
/dev/hda2   /              ext2  defaults          1 1
```

```
/dev/hda1    swap            swap  defaults          0 0
/dev/fd0     /mnt/floopy     ext2  noauto            0 0
/dev/hda3    /users          ext2  defaults,usrquota,grpquota  1 1
none         /proc           proc  defaults          0 0
```

Se creará un archivo y se le dará el permiso correspondiente donde debe guardar los registros de los usuarios

```
#touch /users/quota.user
#chmod 600 /users/quota.user
```

Se reinicia la computadora, se lee este archivo y se empieza el comando ***quotaon***

Después de reiniciar la computadora se puede asignar el espacio al usuario con ***edquota:***

```
#edquota -u login
Quotas for user login:
/dev/hda3: blocks in use: 3872, limits(soft=0, hard=0)
     inodes in use: 375, limits(soft = 0, hard = 0)
```

Donde la información está como:

I) La información del usuario se encuentra en una partición.
II) El número de bloques usados.
III) Los límites de cuota inferiores y superiores (*soft y hard*, respectivamente).
IV) El mismo formato pero ahora para los *inodos.*

Para comprobar o para que el usuario vea su cuota, se utiliza el comando ***quota -v login*** para ver el espacio ocupado del usuario:

```
#quota -v login
```

Con lo anterior, es muy fácil darle cuota a los usuarios y mantener controlado los archivos de los usuarios.

Misceláneas. Con el comando ***quota -v login*** se ve a un usuario en particular, pero si se quiere ver a todos los usuarios se usa ***repquota -a:***

```
#repquota -a
```

También se puede actualizar el archivo */users/quota.user*, cuando se da de alta a un usuario con **quotacheck -a.**

```
#quotacheck -a
```

Ahora se repite el comando ***repquota:***

```
#repquota -a
```

2.3 ARCHIVOS DE DISPOSITIVOS

Todos los periféricos conectados a la computadora que se usan para correr Linux son tratados como archivos especiales (device files) por el sistema operativo. Un periférico o dispositivo es una terminal, un disco duro, una impresora, un manejador de CD-ROM, o un modem. Todo lo que recibe o envía datos hacia el sistema operativo es un dispositivo.

El hecho de tratar todo en el sistema como un dispositivo es uno de los beneficios de la arquitectura Unix. Cada dispositivo tiene un archivo especial llamado manejador de dispositivo (device file), el cual incluye todas las instrucciones necesarias para que Linux se comunique con el dispositivo. Cuando un nuevo dispositivo es desarrollado, puede ser usado por Linux escribiendo su manejador de dispositivo, el cual es usualmente un conjunto de instrucciones que explican como mandar y recibir datos.

Los manejadores de dispositivos permiten al kernel de Linux incluir sólo el sistema operativo y el soporte de software. Teniendo las instrucciones para comunicarse hacia los dispositivos dentro de un conjunto de archivos, estos pueden ser buscados cuando se necesitan (en el caso de que raramente sean usados) o almacenados en memoria todo el tiempo cuando el sistema operativo es reiniciado. Los refinamientos hechos hacia un periférico, pequeños cambios hacia el archivo manejador del dispositivo pueden tener informado al sistema operativo sobre nuevas características y capacidades.

Cuando una aplicación envía datos a un dispositivo, el kernel de Linux no tiene que preocuparse por el mecanismo. El kernel sólo pasa la petición al manejador del dispositivo y deja que éste maneje las comunicaciones. Similarmente, cuando estamos tecleando, el manejador de dispositivo de la terminal acepta la acción y la pasa al shell, filtrando cualquier código especial que el kernel no conozca y lo traduce a un formato que el kernel pueda operar.

Por omisión y convención Linux guarda los manejadores de dispositivos en el directorio `/dev`. Pueden guardarse los manejadores de dispositivos en cualquier parte del sistema de archivos, pero guardándolos en `/dev` hace obvio que son los manejadores de dispositivos.
Diferencia entre dispositivos modo bloque y modo caracter.

Cada tipo de dispositivo en el sistema Linux se comunica con la aplicación en una de las formas siguientes: caracter por caracter o como un conjunto de datos en un bloque de tamaño predefinido. Las terminales, impresoras y módems asíncronos son dispositivos modo carácter. Cuando se usa el modo caracter se envía uno a la vez y hace eco en la otra terminal. Los manejadores (device drivers) de disco duro y la mayoría de manejadores, usan el modo bloque, porque éste es el camino más rápido para enviar o recibir grandes cantidades de información.

Los archivos de dispositivo (device files) son llamados dispositivos modo carácter o dispositivos modo bloque, basados en la forma de comunicación.
Cabe mencionar que los dispositivos que operan a modo carácter son distintos de los de modo bloque, en el aspecto de como el dispositivo maneja su búfer. Los dispositivos modo

carácter hacen su propio búfer. Los dispositivos modo bloque usualmente se comunican en bloques de 512 o 1024 bytes y el kernel se ocupa del búfer.

Algunos periféricos necesitan usar archivos de dispositivo modo bloque y caracter al mismo tiempo. Los manejadores de dispositivo manejan el modo caracter y el modo bloque a través de dos diferentes archivos de dispositivo. El archivo de dispositivo que se usa depende de cómo la aplicación quiera escribir o leer datos hacia el periférico.

El archivo de dispositivo tiene todos los detalles con respecto a que si el periférico opera a modo caracter o modo bloque. Una manera fácil de saber qué tipo de modo utiliza un periférico es obtener un listado largo del archivo de dispositivo. El listado se obtiene con el comando `'ls -l'` que muestra los permisos, dueño, grupo, etc. del archivo. Si el primer carácter es una `b`, indica que el periférico opera en modo bloque y una `c` indica que el periférico opera en modo carácter.

Por ejemplo, para inspeccionar un dispositivo serial en el sistema:

```
# ls -l  /dev/cua0

crw-rw----   1    root       uucp       5, 64 Jul 17 1994 /dev/cua0
```

Los archivos de dispositivos son usualmente nombrados indicando el tipo de dispositivo que son. La mayoría de terminales, por ejemplo, tienen un archivo de dispositivo con el nombre `tty` seguido por dos o más letras o números, tal como `tty1`, `tty1A`, o `tty04`. Las letras `tty` identifican al archivo como una terminal (tty es por teletype), y los números o letras identifican una terminal específica a la que es referida. Cuando se encuentran en el directorio llamado `/dev`, el nombre completo del archivo de dispositivo se convierte en `/dev/tty01`. El manejador del *mouse* conectado a su computadora se accesa a través del manejador `/dev/mouse`.

Lista de nombres de varios manejadores de dispositivos.

Primer floppy (A:)	`/dev/fd0`
Segundo floppy (B:)	`/dev/fd1`
Primer disco duro (completo)	`/dev/hda`
Primer disco duro, partición primaria 1	`/dev/hda1`
Primer disco duro, partición primaria 2	`/dev/hda2`
Primer disco duro, partición primaria 3	`/dev/hda3`
Primer disco duro, partición primaria 4	`/dev/hda4`
Primer disco duro, partición lógica 1	`/dev/hda5`

Primer disco duro, partición lógica 2	/dev/hda6
...	
Segundo disco duro (completo)	/dev/hdb
Segundo disco duro, partición primaria 1	/dev/hdb1
...	
Primer disco duro SCSI (completo)	/dev/sda
Primer disco duro SCSI, partición primaria 1	/dev/sda1
...	
Segundo disco duro SCSI (completo)	/dev/sdb
Segundo disco duro, partición primaria 1	/dev/sdb1
...	

Algunos archivos de los manejadores de dispositivos pueden existir aunque estos no estén instalados. Así que si se tiene el archivo /dev/sda, no significa que realmente se tenga un disco duro SCSI.

Número mayor y menor de dispositivo (Major and Minor Numbers).

Linux puede usar el mismo manejador de dispositivo para todos los dispositivos del mismo tipo. Por ejemplo, un sistema Linux podría tener una tarjeta multipuerto (múltiples puertos seriales) con 10 terminales Wyse 60. Linux puede usar el mismo manejador de dispositivo para cada terminal porque todas ellas son del mismo tipo.

Sin embargo, el sistema operativo tiene un método para diferenciar a cada una de las 10 terminales que se quiera direccionar. Aquí es donde los números de dispositivos son usados. Cada dispositivo es definido por dos números de dispositivo:

a) El número mayor, que identifica el manejador de dispositivo que se ocupara y
b) El número menor, que identifica el número de dispositivo.

Por ejemplo, las 10 terminales Wyse 60 sobre la tarjeta multipuerto pueden usar un manejador de dispositivo con el mismo número mayor (el cual realmente apunta a la localización del archivo manejador de dispositivo en el directorio /dev), pero cada uno tiene un número menor diferente, este es utilizado por el sistema operativo únicamente para identificar la terminal en la tarjeta.

Cada dispositivo en el sistema operativo tiene un número mayor y un número menor de dispositivo asignado único. Si dos dispositivos tuvieran el mismo número, Linux no puede comunicarse propiamente con ellos.

Los números de dispositivo son creados con el comando `mknod` (make node) y se borran con el comando estándar `rm`.

Para el uso del *mouse* se necesitan los siguientes archivos de dispositivo, si no se cuenta con estos archivos de dispositivo, se deberán crear:

Logitech	/dev/logibm	10	0	mknod /dev/logibm c 10 0
PS/2	/dev/psaux	10	1	mknod /dev/psaux c 10 1
Inport	/dev/inportbm	10	2	mknod /dev/inportbm c 10 2
ATI-XL	/dev/atibm	10	3	mknod /dev/atibm c 10 3

Realizar el siguiente ejercicio para crear los diferentes manejadores de dispositivo para el mouse:

```
#cd /dev                          Se cambiará al directorio de manejadores de
                                  dispositivo donde se va a trabajar.
#
#ls -li logibm                    Hace un listado largo con su respectivo número
                                  de inodo del manejador de dispositivo.
12386  crw-rw-r--  1  root  root  10,  0  Dec31  1979
logibm
#
#rm logibm                        Borra el manejador de dispositivo
#
#mknod /dev/logibm c 10 0         Crea un nuevo manejador de dispositivo.
#
#ls -li logibm                    Lista el nuevo manejador para comprobar
                                  que fue creado en forma correcta.
13490  crw-r--r--  1  root  root  10,  0  Nov6  21:26
logibm
#
#chmod 664 logibm                 Cambiar los permisos del archivo para
                                  dejarlo en su forma original.
#ls -li logibm                    Listar el manejador para ver los cambios hechos.
13490  crw-rw-r--  1  root  root  10,  0  Nov6  21:26
logibm
#
```

```
#ls -li psaux                Hace un listado largo con su respectivo
                             número de inodo del manejador de dispositivo.
12421  crw-rw-r--  1  root  root  10, 1  Dec31  1979   psaux
#
#rm psaux                    Borra el manejador de dispositivo
#
#mknod /dev/psaux c 10 1     Crea un nuevo manejador de dispositivo.
#
#ls -li psaux                Lista el nuevo manejador para
                             comprobar que fué creado en forma correcta.
13491  crw-r--r--  1  root  root  10, 1  Nov6  21:31   psaux
#
#chmod 664 psaux             Cambia los permisos del archivo para
                             dejarlo en su forma original.
#
#ls -li psaux                Lista el manejador para ver los cambios hechos.
13491  crw-rw-r--  1  root  root  10, 1  Nov6  21:31   psaux
#

#ls -li inportbm     Hace un listado largo con su respectivo número de inodo
                     del manejador de dispositivo.
12383  crw-rw-r--  1  root  root  10, 2  Dec31  1979
inportbm
#
#rm inportbm                 Borra el manejador de dispositivo
#
#mknod /dev/inportbm c 10 2  Crea un nuevo manejador de dispositivo.
#
#ls -li inportbm             Lista el nuevo manejador para comprobar
                             que fué creado en forma correcta.
13492  crw-r--r--  1  root  root  10, 2  Nov6  21:33
inportbm
#
#chmod 664 inportbm          Cambia los permisos del archivo para
                             dejarlo en su forma original.
#
#ls -li inportbm             Lista el manejador para ver los cambios hechos.
13492  crw-rw-r--  1  root  root  10, 2  Nov6  21:33
inportbm
#

#ls -li atibm                Hace un listado largo con su respectivo
                             número de inodo del manejador de dispositivo.
12210  crw-rw-r--  1  root  root  10, 3  Dec31  1979   atibm
#
```

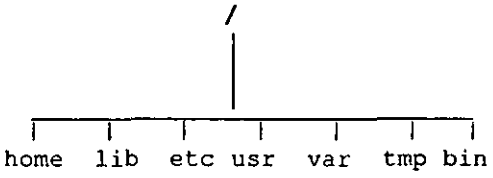
```
#rm atibm                    Borra el manejador de dispositivo.
#
#mknod /dev/atibm c 10 3     Crea un nuevo manejador de dispositivo.
#
#ls -li atibm                Lista el nuevo manejador para comprobar que
                             fué creado en forma correcta.
13489  crw-r--r--  1  root  root  10, 3  Nov6  21:12  atibm
#
#chmod 664 atibm             Cambia los permisos del archivo para
                             dejarlo en su forma original.
#
#ls -li atibm                Lista el manejador para ver los cambios hechos.
13489  crw-rw-r--  1  root  root  10, 3  Nov6  21:12  atibm
#
```

2.4 CONCEPTOS DEL SISTEMA DE ARCHIVOS

2.4.1 El sistema de archivos

El sistema de archivos cuenta con las siguientes características:

Una estructura jerárquica arborescente, en donde el nodo principal es el directorio llamado raíz (representado por "/") y cada uno de los niveles del árbol representa directorios.



Consistencia en el manejo de archivos.
Protección para los datos de los archivos.
Manejo de los dispositivos periféricos como archivos.

La estructura del sistema de archivos consiste en el bloque de boot, el super bloque, la lista de i-nodos y el bloque de datos(17).

1. El *bloque de boot* ocupa la parte del principio del sistema de archivo, por lo regular el primer sector, y puede contener el código de boot o de arranque.

2. El *superbloque* describe el estado de un sistema de archivos. Contiene información acerca del tamaño de los sistemas de archivos, la lista de bloques libres disponibles, índice del siguiente bloque libre en la lista de los mismos, el tamaño, el total y la lista de los i-nodos, el índice del siguiente i-nodo libre, los campos de bloqueo de elementos de las listas de bloques libres y de i-nodos libres. Estos campos se emplean cuando se realiza una petición de bloque o de i-nodo libre, también contiene el *flag* que sirve para indicar si el superbloque ha sido modificado o no.

3. La *lista de i-nodos*, se encuentra a continuación del superbloque. En esta lista, durante el proceso de arranque del sistema, el kernel lee la lista de inodos del disco y carga una copia en memoria, conocida como tabla de i-nodos. Las manipulaciones que hace el subsistema de archivos involucran a dicha tabla, pero no a la lista de i-nodos. Mediante este mecanismo se consigue una mayor velocidad de acceso a los archivos, ya que la tabla se encuentra cargada siempre en la memoria. Los campos de un i-nodo son los siguientes:

a.	Identificador del propietario del archivo
b.	Tipo de archivo
c.	Tipo de acceso al archivo
d.	Tiempos de acceso al archivo
e.	Número de enlaces del archivo
f.	Entradas para los bloques de dirección
g.	Tamaño del archivo

La tabla de i-nodos contiene la misma información que la lista de i-nodos, además de la siguiente información adicional:

I. El estado del i-nodo, el cual indica si está bloqueado, si hay algún proceso esperando a que el i-nodo se desbloquee, si la copia del i-nodo que hay en memoria difiere de la que está en el disco, si la copia de los datos del archivo que existe en memoria difiere de los datos que hay en el disco.
II. El número de i-nodo, como los i-nodos se almacenan en disco en un arreglo lineal, al cargarlo en memoria, el kernel le asigna un número en función de su posición en el arreglo.
III. Apuntadores a otros i-nodos cargados en memoria, el kernel enlaza los i-nodos sobre una cola hash (cola de dispersión) y sobre una lista libre. Las claves de acceso a la cola hash la da el número de dispositivo lógico del i-nodo y el número de i-nodo.
IV. Un contador que indica el número de copias del i-nodo que están activas.

4. El *bloque de datos* empieza a continuación de la lista i-nodos y ocupa el resto del sistema de archivos. En esta zona es donde se encuentra situado el contenido de los archivos a los que se hace referencia la lista de i-nodos. Cada uno de los bloques destinados a datos sólo puede ser asignado a un archivo, tanto si lo ocupa totalmente como si no lo ocupara.

Super bloque	Descripción del FS	Bloque de Bitmap	Bitmap del inodo	Tabla del inodo	Bloque de Datos

Se tienen diseñados e implantados dos nuevos sistemas de archivos y estos están incluidos en el estándar del kernel de Linux. Estos dos llamados "*Extended File System [extfs]*" y "*Second Extended File System [ext2fs]*" arreglan las limitaciones y nuevas características.

Presentaremos la implementación del Sistema de Archivo Virtual en Linux y se detallará el Second Extended File System del código del kernel y las utilerías de modo usuario.

Después de la integración del VFS en el kernel, un nuevo sistema de archivos llamado "Extended File System" fue implementado en abril de 1992 y agregado a Linux 0.96c. Este nuevo sistema de archivos remueve a las 2 grandes limitaciones de Minix. Máximo tamaño fue de 2 Gigabyte y el máximo nombre del archivo fue de 255 caracteres. Fue un mejoramiento sobre el sistema de archivos de Minix pero algunos problemas están presentes. Estos no soportan los accesos separados, modificando el inodo, y las modificaciones de datos están en tiempos estampados. El sistema de archivos usan ligas de listas para guardar "track" de los bloques libres e inodos y estos producen malos performances : como los sistemas de archivos fueron usados, las listas comenzaron sin orden y los sistemas comienzan a fragmentar. Como una respuesta a estos problemas, dos nuevos sistemas de archivos fueron desplegados en la versión alfa de enero de 1993 : el sistema de archivos Xia y el Second Extended File System. El sistema de archivos Xia fue basado en el sistema de archivos Minix y el código del kernel únicamente agrega un poco improvisado sobre los sistemas de archivos. Básicamente, provisto de nombre largo en los archivos, soporta grandes particiones desde 3 tiempos estampados. Por el otro lado, ext2fs fue basado con el código de extfs con mucha reorganización y mejoramiento. Tiene que estar diseñado con evolución en mente y contiene espacio a futuros mejoramientos. Este será descrito con mas detalle en páginas posteriores.

Cuando los dos nuevos sistemas de archivos fueron implantados, provistos esencialmente del mismo aspecto. Estos sistemas de archivos diseñados, Xiafs fue más estable que ext2fs. Como los sistemas de archivos fueron usados más, los bugs fueron fijos en el ext2fs y las mejora tienen nuevos aspectos integrados. Ext2fs es ahora muy estable y tiene al inicio el factor estándar del sistema de archivos de Linux.

Esta tabla contiene un sumario de los aspectos provistos de los diferentes sistemas de archivos

	MINIX FS	EXT FS	EXT2 FS	XIA FS
Max file size	64MB	2GB	2GB	64MB
Max file name	16/30c	255c	255c	248c
3 time support	No	No	Yes	Yes
Extensible	No	No	Yes	No
Var. block size	No	No	Yes	No
Maintained	Yes	No	Yes	?

2.4.2 El sistema de archivos virtual (VFS)

El kernel de Linux contiene un VFS que es usado cuando inicia kernel y durante la llamada al sistema accionando en archivos. El VFS es una capa indirecta la cual tiene un archivo orientado a la llamada del sistema y llama necesariamente funciones en el sistema de archivos físicos codificados en el I/O. Este mecanismo indirecto es frecuentemente usado en los sistemas operativos UNIX, fácilmente a la integración y el uso de varios tipos del sistema de archivos. Cuando un proceso saliente a un archivo orientado a la llamada del sistema, la llamada del kernel como función contiene el VFS. Esta función contiene una estructura independiente manipulando y redireccionando la llamada a una función contenida en el código del sistema de archivos físicos, el cual es la responsable de la cabecera a la estructura que depende de operaciones. El código de sistemas de archivos usa el buffer cache correspondiendo a las funciones de I/O a los dispositivos.

La definición del VFS pone las funciones como un buen sistema de archivos que está implantando. Esta interfase hace una puesta de operaciones asociadas a tres grandes objetos: sistemas de archivos, inodos y archivos abiertos. El conocido VFS acerca de los tipos de sistema de archivos soportados en el kernel. Usa una tabla definida durante la configuración del kernel. Cada entrada de esta tabla describe un tipo de sistema de archivos: Contiene el nombre del tipo del sistema de archivos y que pone en una llamada de función durante el montaje de operación. Cuando un sistema de archivos es montado, el montaje apropiado de la función es la llamada al sistema. Esta función es responsable de leer el superbloque desde el disco, inicialmente variables internas y retornar un montaje de los sistemas de archivos describe el VFS. Después de que el sistema de archivos es montado, las funciones del VFS puede usar una descripción para acceder a las rutinas físicas de los sistemas de archivos.

Un sistema de archivos montado describe el contenido de algunos datos: información que es común son los tipos de sistemas de archivos, punteando funciones provistas por los sistemas de archivos físicos en el código del kernel, y datos privados mantenidos por el código físico de los sistemas de archivos. La función puntera contiene la descripción del sistema de archivos alojados en el VFS a accesar la rutinas internas del sistemas de archivos.

2.4.3 El segundo sistema de archivo extendido

El Second Extended File System (ext2fs) ha sido diseñado e implantado para fijar algunos problemas presentados en el primer extfs.

En el sistema operativo, necesitamos que el ext2fs tuviera un excelente performance. Se necesita que fuera un muy robusto sistema de archivos en orden a reducir los riesgos de datos perdidos en uso intensivo. Lo último es que el ext2fs tiene que incluir provisiones de extensiones para alojar usuarios o beneficiar desde nuevos aspectos sin reformas el sistema de archivos.

2.4.4 Reformas del estándar ext2fs

Soporta el ext2fs los estándares de los tipos de archivos UNIX : archivos regulares, directorios, archivos especiales de dispositivos, ligas, sockets, etc. Ext2fs es capaz de manejar sistemas de archivos creados en grandes particiones. El código del kernel original restringe el máximo tamaño de los sistemas de archivos a 2 GB. Recientemente el trabajo en la capa de VFS se tiene un limite de 4 TB. Es posible usar discos grandes sin necesidad de crear muchas particiones.

Ext2fs proviene de la longitud de los nombres de archivos. Usa longitud variable en la entrada de directorios. El máximo tamaño del nombre del archivo es de 255 caracteres. Este limite puede ser extendido a 1012 si es necesario. Ext2fs reserva algunos bloques del superusuario (root). Normalmente es del 5% los bloques reservados. Esto aloja al administrador a recobrar fácilmente situaciones cuando los procesos llenan el sistema de archivos.

2.4.5 Reformas avanzadas del ext2fs

En suma el aspecto estándar UNIX, ext2fs soporta algunas extensiones que no se presentan en los sistemas de archivos UNIX. Atribuidos al archivo obtenga del usuario a modificar el kernel cuando acciona una puesta de archivos. Uno puede poner atributos a un archivo o a un directorio. En el último de los casos, nuevos archivos creados en el directorio inherente a estos atributos. Una opción de montaje puede alojar al administrador a escoger la semántica de la creación de archivos. En un sistema de archivos montados con semánticas BSD, son archivos creados con el mismo grupo *id* como es el directorio padre. System V usa en su semántica un bit más complejo: Si un directorio tiene puesto un *bit setgid* en nuevos archivos inherentes al grupo *id* del directorio y subdirectorios inherentes al grupo *id* y el *setgid*; en el otro caso, archivos y subdirectorios son creados con el grupo primario *id* de los llamados procesos con el grupo primario *id* de los llamados procesos.

Una opción de montaje aloja al administrador a solicitar este metadato (inodos, bloques de bitmap, bloques indirectos y directos) deben ser descritos y sincronizados en el disco y modificándolos. Esto puede ser provechoso a mantener un estricto metadato consistente, pero esto liderean una pobre ejecución. Actualmente, este aspecto no es normalmente usado, desde entonces una suma al ejecutar una asociación pérdida con actualizaciones y se usan la sincronización de los metadatos, esto puede causar corrupción en el uso de los datos cual no es una bandera por la cual el filesystem es checado.

Ext2fs aloja al administrador a escoger el bloque lógico de tamaño cuando crea el sistema de archivos. El tamaño de bloques puede ser típicamente de 1024, 2048 y 4096 bytes. Usando grandes tamaños de bloques se pueden correr los I/O desde pocas I/O de respuestas, y así pocos discos de cabeceras, la cual necesitamos al acceder un archivo. En la otra cabeza grandes bloques libres y más espacio en disco: En el promedio, el último bloque alojado a un archivo esta medio lleno, así como bloques obtenidos con más espacio

espera en el último bloque de cada archivo. En suma, la ventaja de los tamaños de bloques largos que son obtenidos por el sistema de archivos con técnicas de prealojamiento.

La implantación de ext2fs de rápidas ligas simbólicas. Una liga simbólica rápida no es usada en un bloque de datos en el sistema de archivos. El nombre en blanco no es almacenado en un bloque de datos en el inodo. Esta política nos salva de algún espacio en el disco (no a bloques de datos que necesitamos que sean alojados) y velocidades arriba de las operaciones ligadas (Esto no es necesario para leer en un bloque de datos cuando se accede mucho a una liga), el espacio disponible en el inodo es limitado y no esta ligado o puede ser implantado como una liga simbólica. El tamaño máximo del nombre en blanco es un símbolo de ligado rápido de 60 caracteres, planeamos a extender este esquema de archivos a pequeños archivos en un futuro cercano.

Ext2fs guarda el track en el estado de los sistemas de archivos. Un campo especial en el superbloque es usado por el código del kernel que indica el estado del sistema de archivos. Cuando un sistema de archivos es montado en modo de lectura/escritura , este modo está puesto ó indicado como "no limpio". Cuando es desmontado o remontado en un inodo en un modo solamente lectura, este estado es reseteado como "limpio". Como en el tiempo de *boot,* el sistema de archivos checa usando esta información y decide si el sistema de archivos es verificado. El código del kernel también graba errores en estos campos. Cuando una inconsistencia es detectada por el código del kernel, el sistema de archivos es marcado como "erróneo". El sistema de archivos verifica con un test y forza la verificación del sistema de archivos de estos aparentes estados "limpios".

Siempre que el sistema de archivos salta a verificar en algún tiempo si esta dañado, así el ext2fs provee 2 caminos para hacer la verificación y forzando en intervalos regulares. Un contador de montaje es mantenido en el superbloque. Cada tiempo el filesystem es montado en modo lectura/escritura, este contador es incrementado. Cuando este valor esta al máximo (también grabado en el superbloque), el filesystem se verifica y hace que el filesystem este "limpio". Una última verificación de tiempo y un máximo al verificar el intervalo que son también mantenidos en el superbloque.

Estos 2 campos alojados al administrador solicitando la verificación periódicamente. Cuando el máximo chequeo del intervalo ha sido alargado, el verificador ignora el estado del filesystem y lo checa en el mismo momento que lo detecta. Ext2fs ofrece utilerías al comportamiento del filesystem. El programa ***tunefs*** puede ser usado para modificar.

I.El error de comportamiento. Cuando una inconsistencia es detectada por el código del kernel, el filesystem es marcado como "erróneo" y una de las 3 siguientes acciones puede ser acabada :

- continúa una normal ejecución, remonta el filesystem en modo solamente lectura para evitar que el filesystem este corrupto.
- hacer pánico en el kernel y rebotear.
- correr el chequeo del filesystem.

II.El contador máximo de montaje.
III.El chequeo máximo del intervalo.
IV.El número de bloques lógicos reservados para el superusuario.

Las opciones de montaje pueden también ser usadas para cambiar el comportamiento erróneo del kernel. Un atributo alojado al usuario que solicita seguridad al borrar los archivos. Cuando un archivo es borrado, el dato aleatorio está escrito en el bloque del disco previo alojado en el archivo. Esta prevención maliciosa de gente desde la ganancia en el acceso previo al contenido de archivo usando un editor de disco.

Ultimo, nuevos tipos de archivos inspirados desde la versión 4.4 BSD del filesystem que está recientemente agregado a ext2fs. Archivos inmutables pueden únicamente ser leídos: nadie puede escribir o borrar esto. Esto puede ser usado para proteger configuraciones de archivos sensitivos. Archivos únicamente agregados puede ser abiertos en modo escritura pero los datos está siempre agregando al final del archivo. Archivos similarmente inmutables, estos no pueden ser borrados o renombrados. Esto es especialmente útil a los archivos, los cuales pueden únicamente crecer(21).

2.5 CREANDO Y USANDO EL SISTEMA DE ARCHIVOS

2.5.1 Introducción

La parte más importante de cualquier sistema de cómputo son los datos, es decir, la información que los programas almacenan y manipulan. Linux tiene un sistema de archivos cuyo trabajo es conservar toda la información que se almacene en la computadora, incluyendo programas, documentos, bases de datos, textos, etcétera.

En Linux el término "archivo" se refiere a cualquier fuente de entrada o destino de salida, no sólo a un depósito de datos. El espacio visible a los usuarios se basa en una estructura de árbol, con la raíz en lo alto. Los archivos y los directorios se colocan debajo de la raíz.

En realidad, algunos de los directorios en el árbol de archivos están físicamente ubicados en diferentes particiones del disco, sobre diferentes discos y eventualmente en diferentes computadoras. Cuando una de estas particiones del disco está ligada con el árbol de archivos en el directorio conocido como punto de montaje (*mount point*), al punto de montaje y a todos los directorios inferiores nos referiremos como el **SISTEMA DE ARCHIVOS.**

Linux tiene ocho tipos de archivos y hablaremos de tres: los ordinarios, los directorios y los especiales. Los archivos ordinarios contienen datos y se almacenan en un disco. Un directorio se almacena en disco y contiene información que se usa para organizar y permitir el acceso a otros archivos. Los archivos especiales o de dispositivo son la representación interna de un dispositivo físico(12).

2.5.2 Montaje de dispositivos

Para poder montar el sistema de archivos en Linux, primero debemos tener una partición física en el disco, un CD-ROM, o un floppy.

Linux usa el comando 'mount' para montar el sistema de archivos. Las formas más utilizadas son:

```
/bin/mount [opciones] dispositivo  punto-de-montaje
/bin/mount punto-de-montaje
/bin/mount -at nfs
```

Donde las *opciones* son las banderas que soporta el comando mount, *dispositivo* es el nombre del archivo de dispositivo (device file) de modo bloque que se desea montar y *punto-de-montaje* es el directorio donde se montará el sistema de archivos. Cabe mencionar que el *punto-de-montaje* es un subdirectorio que debe estar vacío, si este subdirectorio tiene alguna información; ésta quedará oculta cuando se monte el sistema de archivos. El comando tiene varias banderas, estas son:

-f	Termina todos los procesos excepto la actual llamada al sistema.
-v	`mount` provee información adicional sobre lo que se trata de hacer.
-w	El sistema de archivos se montará con permisos de lectura y escritura.
-r	El sistema se montará con permisos de solo lectura.
-n	Se monta sin escribir las entradas en el archivo `/etc/mtab`
-t tipo	Especifica el tipo de sistema de archivos por instalar. Los tipos validos son: `minix,ext,ext2,xiafs,msdos,hpfs,proc,nfs,` `umsdos,sysv` e iso9660.
-a	Intenta montar todo el sistema de archivos, descritos en `/etc/fstab`
-o lista de opciones	El argumento `-o` le dice a `mount` que aplique las opciones que son especificadas para el sistema de archivos que se montará. Para más detalles ver la página de ayuda con el comando 'man mount'

Realice los siguientes pasos:

1.- Se introduce un disco flexible en la unidad de disco.
2.- Se ejecuta el comando:

```
#/sbin/mount -o conv=auto -t msdos /dev/fd0 /mnt
```

Este tipo de montajes se pueden realizar automáticamente, cuando el sistema entra a modo multiusuario y se edita el archivo de configuración llamado: `/etc/fstab`.

Un archivo común es como el siguiente:

```
#
# /etc/fstab
#
# You should be using fstool (control-panel) to edit this!
#
# <device>    <mountpoint>   <filesystemtype> <options> <dump>
<fsckorder>

/dev/hda2          /                         ext2    defaults 1 1
/dev/hdb1          /disco                    ext2    defaults 0 0
/dev/fd0           /mnt/floppy               ext2    defaults,users,noauto
0 0
/dev/hda1          none                      msdos   defaults 0 0

/proc              /proc                     proc    defaults
/dev/hda3          none                      swap    sw

cronos:/reas       /users                    nfs     rw
brahm:/u/alum      /users/alum1              nfs     rw
brahm:/u2/alum     /users/alum2              nfs     rw
```

2.5.3 Desmontaje de dispositivos

El proceso contrario de montar es **desmontar** un sistema de archivos. Al igual que con el comando `'mount'` tenemos tres distintas formas muy usuales del comando `'umount'`, estas son:

```
#/bin/umount dispositivo | punto-de-montaje
#/bin/umount -a
#/bin/umount -t fstipo
```

donde *dispositivo* es el nombre del dispositivo físico por desmontar y *punto-de-montaje* es el nombre del directorio donde fué montado. Solamente se necesita especificar una u otra opción. El comando tiene dos parámetros adicionales: `-a` desmonta todo el sistema de archivos, y `-t fs-tipo` actúa sólo en los sistemas de archivos especificados por fs-tipo.

```
#/bin/umount /dev/fd0
#/bin/umount /mnt
```

2.5.4 formato de un disco (Fdisk)

Cuando se conecta un nuevo disco duro a la computadora, se necesita particionar el disco a fin de poder montar un sistema de archivos en él. La partición se realiza con el comando `'fdisk'`.

Para una referencia detallada de este comando se puede revisar el manual de Linux sobre 'fdisk'.

```
# fdisk drive
# fdisk /dev/hdb
```

Las opciones anteriores son típicas para el comando.

2.5.5 Creación del sistema de archivos con mkfs

Una vez que se ha realizado una partición en el disco con el comando 'fdisk', se debe crear el sistema de archivos, antes de proceder a ocupar el disco para introducir los datos. Esto se puede realizar con el comando 'mkfs'. La forma de usar el comando es la siguiente:
mkfs [-V] [-t *fs-tipo*] *[fs-opciones] sist_archivo* [*bloques*]

sist_archivo	Es el nombre especial del sistema de archivos que se desea construir tal como /dev/hda1.
-V	Especificando esta opción inhibe más de una ejecución del sistema de archivos.
-v	*verbose*. Información extra.
-t fs-tipo	Especifica el tipo de sistema de archivo a construir. Ver el manual en línea de: fsck(8), mkfs.minix(8), mkfs.ext(8), mkfs.ext2(8), mkfs.xiafs(8). El tipo por omisión es minix.
fs-opciones	Las opciones con las que se creará el actual sistema de archivos.
-c	Verifica el dispositivo para los bloques malos antes de construir el archivo.
Bloques	Especifica el número de bloques a usar por el sistema de archivos.

Un ejercicio práctico se realiza con discos flexibles utilizando el siguiente comando:

```
# /sbin/mkfs -t ext2 /dev/fd0 1440
```

2.5.6 Verificando el sistema de archivos con fsck

En muchas ocasiones el sistema de archivos se daña. La causa más común es apagar la PC sin antes haber desmontado los sistemas de archivos. En los sistemas UNIX se ofrece el comando 'fsck' (que a su vez es una liga a *e2fsck*)para reparar el sistema de archivos dañado. En Linux el comando tiene la siguiente sintaxis:

fsck [-A] [-V] [-t *fs-tipo*] [-a] [-l] [-r] [-s] *sis_arch*

-A	Va a través del archivo /etc/fstab y trata de checar todo el sistema de archivos en una pasada.
-V	Imprime información adicional acerca de lo que 'fsck' va haciendo.
-t fs-tipo	Especifica el tipo de sistema de archivo por verificar.
-a	Automáticamente repara cualquier problema que encuentra en el sistema de archivos sin preguntar. *Use esta opción con cuidado*
-l	Lista todos los nombres de archivos en el sistema de archivos.
-r	Pregunta la confirmación antes de reparar el sistema de archivos.
-s	Lista el superblock antes de checar el sistema de archivos.
Sis_arch	Especifica el sistema de archivos que será verificado.

El programa *e2fsck* está diseñado para correr rápidamente si es posible. Desde la verificación del filesystem tiene que ser un disco configurado, esto es para optimizar los algoritmos usados en el *e2fsck y se* esta estructurando el filesystem y no son repetidos para acceder desde el disco. En suma, el orden en el cual los inodos y los directorios son verificados, ordenados por el número de bloques reducidos al importar el tiempo de discos buscados. Algunas de estas ideas son originalmente exploradas por [Bina y Emrath 1989] por esto tiene que ser refinados por los autores.

En el paso1, *e2fsck* itera sobre todos los inodos en los filesystem y verifica sobre cada inodo con una no conectividad de objeto en el filesystem. Esto es, la verificación no requieren chequeos de los objetos del filesystem. Los ejemplos de cada verificación incluyen hechos seguros del archivo de modo legal, y esto es en todos los bloques en el inodo son número de bloques válidos. Durante el paso 1, los bitmaps indican cuál bloque e inodo están en uso y que están compilados.

Si *e2fsck* notifica datos de bloques los cuales son reclamados por más de un inodo, invoca el paso 1B por 1D que resuelve estos conflictos, uno a otro y clona los bloques compartidos, así cada inodo fue copiado por el bloque compartido por desalojados de uno o más de los inodos.

Paso 1 toma la longitud de tiempo por ejecutar, desde todos los inodos tiene que ser leído dentro de la memoria y checado. Reduce el tiempo de I/O necesario en futuros pases, la información del filesystem crítico está intercambiando en memoria. El mas importante ejemplo de esta técnica es la localización en el disco de todos los bloques en el filesystem. Esto obviamente necesita la relectura de la estructura de los inodos de directorios durante el paso 2 que contiene esta información.

El Paso 2 revisa directorios como objetos sin conexión. Desde el directorio de entrada no resuelve bloques de disco, cada bloque de directorio puede revisarse individualmente sin referencia a otros bloques. Esto implica que el *e2fsck* ordena todos los bloques de directorios por el número de estos y revisa el directorio de bloques en orden ascendente, así decrece la búsqueda del disco. Los bloques de directorios son revisados para asegurar las entradas de directorios que son válidos, y contiene la referencia a los números de los inodos que están en uso (como se determinan por el paso 1).

Desde el primer directorio de bloque en cada inodo del directorio, el "." y ".." de entrada son revisados asegurando, si existen y el número del inodo de la entrada del "." Observando en el directorio corriente el número del inodo de la entrada ".." no es revisado hasta el paso 3. El paso 2 también revisa la información concierte al directorio padre en el cual cada directorio es ligado. Si un directorio es referente por más que en un directorio, la segunda referencia del directorio es negociado como una ilegal liga dura y es removida. Es importante hacer notar que el fin del paso 2, casi todos los I/O del disco el comando *e2fsck* necesita ejecutarse completo. Información de los pasos 3, 4 y 5 son intercambiados en memoria ; De aquí los pasos quedan en *e2fsck* y son largamente atados al CPU, además se dan pérdidas que están al 5-10 % del tiempo total que corre el *e2fsck.*

En el paso 3, la conectividad del directorio es revisado. Los trazos *e2fsck* del ruteo de cada directorio atrás de la raíz, usando la información que fue intercambiada durante el paso 2. Como estos tiempos, la entrada ".." de cada directorio es también checado a hacer seguro que es válido. Algunos directorios los cuales no pueden ser trazados atrás del root son ligados al directorio */lost+found.*

En el paso 4, *e2fsck* revisa la referencia del contador de todos los inodos, por interacción sobre todos los inodos y compara las ligas contadas (lo cual fue intercambiando en el paso 1) fue el contador interno compuesto durante el paso 2 y 3. Algunos archivos sin borrar con un contador de liga cero es también ligado al directorio */lost+found* durante este paso.

Finalmente en paso 5, *e2fsck* revisa la validación del sumario de la información del filesystem. Compara el bloque y el bitmap del inodo que fue construido durante los pasos previos y contra los cuales bitmaps del filesystem, y las copias correctas del disco si es necesario. La forma más común de usar '`fsck`' es la siguiente:

```
# fsck /dev/hda3
```

2.6 MANEJO DEL SISTEMA DE ARCHIVOS

Existen varios tipos de archivos

1. *Archivos ordinarios.* Son los más comunes, los que almacenan datos, es decir, puede ser un programa, un archivo de texto, código fuente o cualquier cosa que pueda guardarse en cualquier lugar. El kernel soporta acceso secuencial y aleatorio en todos estos archivos.

2. *Directorios.* Tienen en común con los archivos ordinarios que también contienen datos, sólo que en este caso el dato es una lista de otros archivos.

3. *Archivos especiales.* Se identifican porque cada uno tiene un número de dispositivo mayor y uno menor (major and minor device number). El número mayor identifica al manejador del dispositivo que necesita el kernel para acceder al dispositivo. El número menor significa un parámetro dependiente del manejador del dispositivo usado típicamente para diferenciar entre diversos tipos de dispositivos soportados por el manejador, o distintos modos de operación. Utilizan un i-nodo pero no bloques de datos. Representan dispositivos en los que se pueden leer o escribir cantidades arbitrarias de datos. Incluyen sistemas de archivos, puertos seriales, puertos paralelos, terminales y cintas. También se les conoce como *"raw devices"* debido a que no manipulan la I/O. Los discos duros y flexibles pueden ser accedidos de esta manera. Podemos ver archivos de este tipo en el directorio */dev.*

4. *Entubamientos* (FIFO [First In First Out]). Son aquellos que permiten la comunicación entre dos procesos ejecutándose en el mismo nodo. Los entubamientos pueden ser creados con el comando `mknod` y eliminados con el comando `rm`.

5. *Ligas duras.* En realidad una liga no es un archivo, es un nombre adicional para otro archivo. Cada archivo tiene al menos una liga, usualmente el nombre bajo el cual fue originalmente creado. Cuando se hace una nueva liga hacia un archivo, un alias para este archivo es creado. Una liga es indistinguible del archivo al cual está ligado; LINUX mantiene el conteo de la cantidad de ligas que apuntan hacia un archivo en particular y no libera el espacio que ocupa el archivo hasta que la última liga es eliminada. La liga dura es una conexión directa entre archivos, por lo que ésta no puede existir a través de distintos sistemas de archivos.

6. *Ligas simbólicas.* Son archivos que simplemente contienen el nombre de otro archivo. Cuando el kernel trata de abrir o pasar a través de la liga, su atención es directamente hacia el archivo que la liga simbólica apunta en vez de abrir la liga simbólica en sí. La diferencia entre las ligas, es que las duras son una referencia directa, mientras que las simbólicas son una referencia a través de un archivo, las simbólicas son el archivo en sí, por lo tanto, tienen un propio dueño y permisos.

7. Sockets. Son conexiones entre procesos que les permiten comunicarse de una manera más rápida y fácil. Existen varios tipos de sockets en UNIX, muchos de los cuales involucran el uso de la red. Los sockets son locales a un modo en particular y son referenciados a través de un objeto en el file system en vez de un puerto en la red. Los archivos de sockets son visibles a los demás procesos como entradas en el directorio, estas entradas no pueden ser leídas o escritas por procesos que no estén involucrados en la conexión del socket(17).

Los sockets en UNIX son creados con la llamada al sistema `socket()` y pueden ser eliminados con el comando `rm` con la llamada al sistema `unlink ()`.
La llamada para abrir un canal bidireccional de comunicaciones es socket y se declara como sigue:

```
#include <sys/typesh.h>
#include <sys/socket.h>

int socket(af,type,protocol)
int af,type, protocol;
```

Socket crea un punto terminal para conectarse a un canal y devuelve un descriptor. El descriptor de socket devuelto se usará en llamadas posteriores a funciones de la interfaz.

Af (address family) especifica la familia de sockets o familia de direcciones que se desea emplear. Las distintas familias están definidas en el archivo de cabecera *<sys/socket.h>* y dependerán del fabricante del sistema y de la configuración del hardware. Las dos familias siguientes suelen estar presentes en todos los sistemas:

`AF_UNIX` Protocolos internos UNIX. Es la familia de sockets empleada para comunicar procesos que se ejecutan en una misma computadora. Esta familia no requiere que esté presente un hardware especial de red, puesto que en realidad no realiza accesos a ninguna red.

`AF_INET` Protocolos Internet. Es la familia de sockets que se comunican mediante protocolos, tales como *TCP (Transmission Control Protocol)*, desarrollado por la Universidad de California en Berkeley para *DARPA (Defense Advance Research Projects Agency)* ó *UDP (User Datagram Protocol).*

`AF_CCINT` Norma X.25 del CCITT.

`AF_NS` Protocolo NS de Xerox..

El argumento *type* indica la semántica de la comunicación para el socket. Puede ser:

`SOCK_STREAM` Socket con un protocolo orientado a conexión. Esto es lo que hemos estudiado como circuito virtual.

SOCK_DGRAM Socket con un protocolo no orientado a conexión o datagrama.

Protocol especifica el protocolo particular que se va usar en el socket. Normalmente, cada tipo de socket tiene sólo un protocolo, pero si hubiera más de uno, se especificaría mediante un argumento. *Protocol* puede valer cero, en cuyo caso la elección del protocolo se deja en manos del sistema. Si la llamada se ejecuta satisfactoriamente, devolverá un descriptor de fichero válido. En caso contrario, devolverá -1 y el *errno* estará codificado al error producido.

Los *Permisos de Archivos* son el conjunto de nueve bits de permisos asociados a ellos, que controlan quien puede leerlos, escribirlos o ejecutarlos. Y junto con otros tres bits que afectan la forma de ejecutarlos, constituyen los llamados *bits de modo*, que son los permisos totales del archivo. Los doce bits de modo son guardados con otros 4 bits tipo de archivoque son fijados cuando el archivo es creado y no pueden ser modificados, pero los otros 12 bits pueden ser alterados por el dueño o por el superusuario usando el comando chmod. El comando `/bin/ls -l` es utilizado para examinar el valor de estos bits.

Los bits correspondientes son el *setuid*, *setgid* y *sticky-bit* y estos valores octales son 4000, 2000 y 1000 respectivamente.

Fijando el bit SUID (Set User ID) en un archivo, permite que los usuarios ejecuten los permisos del dueño del archivo.

Fijando el bit SGID (Set Group ID) en un archivo, permite que los usuarios ejecuten los permisos del grupo del archivo.

El *Sticky bit* (bit pegajoso). Es el bit con valor octal de 1000. Cuando este bit es fijado a un archivo ejecutable, el bit le indica al sistema operativo que el archivo se ejecutará con mucha frecuencia y por lo tanto deberá ser retenido en la memoria principal aún cuando no sea ejecutado. Esto desperdicia memoria principal, pero reduce el tiempo de ejecución del programa significativamente.

Tipo de archivo	Símbolo	Creado por	Eliminado por
Archivo regular	"-"	editores, cp , etc.	rm
Directorio	"d"	mkdir	rmdir, rm -r
Dispositivo de modo carácter	"c"	mknod	rm
Dispositivo de modo bloque	"b"	mknod	rm
Socket	"s"	socket(2)	rm
Named pipe	"P"	mknod	rm
Liga simbólica	"l"	ln -s	rm

2.7 MANEJO DEL AREA DE SWAP

El área de swap también es conocida con el nombre de espacio de intercambio, esta área se utiliza cuando la memoria RAM es limitada, por lo que es necesario utilizar memoria virtual o área de swap.

Linux soporta dos tipos de área de swap: particiones de swap (device swap) y archivos de swap (file system swap). Una partición de swap es una partición física de disco, la cual tiene un ID de sistema de archivos establecida con el número 82, que es el número que identifica el área de swap de Linux. Un archivo de swap es grande en un sistema de archivos, que se emplea como área de swap(18).

2.7.1 Memoria virtual

El manejo de memoria comienza con una emisión compleja de sistemas multiusuarios debido al total de requerimientos de los usuarios y del sistema operativo, generalmente se exceden de la memoria disponible del sistema. Incluso la rápida reducción de precios de *chips* de memoria y consecuentemente el crecimiento en la memoria principal no tiene la capacidad de vencer las necesidades de mayor requerimiento de memoria. El tema comienza con todas las soluciones de mas memoria en el uso de una parte del disco común a una extensión de la memoria principal. Llamada memoria virtual, este almacenamiento de memoria provee espacio para todos los usuarios y puede ser expandida como se necesite si el número de usuarios del sistema esta configurado para crecer. Desde entonces el CPU puede usar únicamente la memoria principal para ejecutar el código y datos de acceso. El código y datos de componentes de un proceso tienen que ser copiado en la memoria principal para ejecución. Entre los esquemas usados de manejo están los siguientes:

1.- *Swapping*
2.- *Demand Paging*
3.- *Demand Paging* y *Swapping*

I. El sistema *Swapping* intercambia, procesos completos (textos, datos, pilas y otras tablas de sistema, necesita la administración de un proceso de tiempo que está en la memoria principal).

Un proceso reside en disco hasta que está listo para correr otra vez. Cuando se hace un área de *swap* en la memoria principal y se etiqueta basado en su prioridad. Entre mas grande sea la memoria principal disponible, será mas grande el número de procesos que estén residentes en memoria.

II. El sistema *Demand Paging*, por otro lado, no intercambia para un proceso completo. El copiado mínimo necesario para correr los procesos y el uso del concepto de fallas de paginas que indican la necesidad de brincar a otra página de los procesos de memoria virtual en disco a memoria física. Una falla de página ocurre cuando la página

requerida no esta físicamente en memoria, y las secciones para leerlo a memoria física que son comenzadas. Cada proceso esta parcialmente en memoria todo el tiempo.

III. El sistema *Demand Paging* y *Swapping* por otro lado, no solo usa únicamente "demanda de página" porque tiene la capacidad de intercambiar procesos completos, si es necesario para hacer espacio de *swapping* parcial en otros procesos. Este esquema se beneficia de ambos, tanto del primero como el segundo esquema.

Es más recomendable utilizar una partición de swap en vez de un archivo, porque todo el acceso a un archivo de este tipo se realiza por medio del sistema de archivos de Linux. Como los bloques de disco son los que conforman el archivo de swap, puede suceder que no estén juntos y, por lo tanto, el rendimiento no será tan bueno como una partición de swap. La E/S con una partición de swap se ejecuta directamente hacia el dispositivo, y los bloques de la partición de disco están siempre juntos, además, al mantener el área de swap fuera del sistema de archivos, reduce el riesgo de corromper el sistema de archivos, si es que algo extraño sucede con el archivo de swap.

2.7.2 El buffer cache

Leer un disco (excepto un disco RAM por razones obvias) es muy lento comparado con el acceso a memoria (real). Además es común leer la misma parte de un disco varias veces durante un tiempo relativamente corto; Por ejemplo, tal vez primero se lea un mensaje de un e-mail, entonces se leerá la carta en un editor cuando se le conteste. Se hará que el programa de correo lo lea otra vez y cuando lo copie hacia un fólder, o considere qué tan seguido se use el comando **ls**, tal vez sea corrido en un sistema con muchos usuarios. Leyendo la información del disco una sola vez y luego guardándola en la memoria hasta que ya no se necesite, se puede aumentar la velocidad como en la primera leída. Esto es llamado *buffering* del disco y la memoria usada para el propósito es llamada el cache del buffer.

Debido a que la memoria es, desafortunadamente finita, el cache del *buffer* usualmente no puede ser suficiente (no puede contener todos los datos que uno quiera usar). Cuando el cache se llena, la información que no ha sido usada por mucho tiempo es descartada y la memoria entonces es usada libremente por la nueva información.

El *buffering* del disco trabaja para escribir también. Por un lado, la información que se escribe pronto se vuelve a leer (por ejemplo un archivo de código fuente es salvado en un archivo, entonces escrito por el compilador), así se pone la información que se escribe en el caché es una buena idea. Por otra parte, solo por poner la información en el cache y no escribirla en el disco de una vez, el programa que se escribe y corre más rápido. Las escrituras pueden llevarse a cabo en *background*, haciendo más lentos otros programas.

Muchos de los sistemas operativos tienen caché del *buffer* (aunque se llamen de otra forma), pero no todos trabajan de acuerdo con los principios arriba mencionado. Algunos son a través de escritura: la información es escrita en el disco una vez (se mantiene en el caché también, desde luego). El caché es llamado escritura-atrás si los escritos son hechos en un momento más tarde, escritura-atrás es más eficiente que a través-escritura, pero también más propensa a errores. Si la computadora se colisiona, o la energía se cortada en un mal momento o el disco flexible es removido del drive del disco antes que la información de datos, los cambios en el cache se pierden. Esto tal vez signifique que el sistema de archivos (si existe) no este en completo orden de trabajo, tal vez porque la información no escrita contenga importantes cambios a la información contenida en libros.

Debido a esto nunca se debe apagar la energía sin usar un procedimiento de apagado apropiado (sistema de arranque y paro) o remover un floppy del drive de disco (montado y desmontado) o después de que el programa que esté usando señale que ha terminado y la luz del drive del disco flexible éste apagado. LINUX tiene un *daemon* llamado **bdflush,** que verifica que la información sea escrita en disco pero éste es un tanto imperfecto como el `sync` que corre en las tradicionales UNIX, donde en éstas se hace un `update` (el comando que realiza la verificación de la información sea escrita en disco) cada 30 segundos, por lo que el `sync` ya no es necesario.

Para crear una partición de swap se requiere crear antes una con el comando `fdisk` (éste comando también nos sirve para verificar el estado actual de la partición del disco) y etiquetarla con el ID 82, para que Linux lo reconozca como anteriormente se mencionó.

El tamaño de la partición depende de la necesidad de qué tanta memoria se requiera como memoria virtual, se tienen que crear múltiples particiones de swap, ahora si el sistema de LINUX lo requiere, se debe crear una partición de swap para el sistema y otra para los archivos del usuario, una sugerencia de algunos autores es la que se llama "Regla del Pulgar", la cual nos indica que para crear el área de swap, tomamos el doble de tamaño de la RAM.

Para *crear la partición formal de swap* se debe de tener la partición ordinaria del disco, después se deben seguir ciertos pasos para hacer la partición activa. El primer paso es crear la partición en forma similar a la de un sistema de archivos, esto es por medio del comando `mkswap`, cuya sintaxis es:

```
# /sbin/mkswap [-c]  dispositivo tamañoenbloques
```

donde dispositivo es el nombre de la partición de intercambio, como `/dev/hda2`, y `tamañoenbloques` es el tamaño del sistema de archivos de destino en bloques. Para tener una mejor idea podemos ejemplificarlo así:

El tamaño de `/dev/hda2` es de 19,159 bloques, ya que Linux requiere que las particiones de área de swap sean de 9 a 65,537 bloques. Se puede obtener el tamaño en bloques al

ejecutar *fdisk* y observar los datos de la tabla de partición . El argumento de *-c* le dice a *mkswap* que revise el sistema de archivos buscando bloques dañados cuando crea el área de swap.

Por lo tanto quedaría:

```
# /sbin/mkswap -c /dev/hda2 19159
```

El siguiente paso después de que se haya creado y preparado la partición, es activarla para que el kernel de Linux la pueda utilizar. El comando para activarla es *swapon*, cuya sintaxis es:

```
# /sbin/swapon sistemaarchivo
```

donde *sistemaarchivo* es el sistema de archivos que se quiere tener disponible como espacio de swap. Linux hace una llamada a *swapon -a* durante el arranque, lo que monta todas las particiones de swap disponibles que están listadas en */etc/fstab*.

Archivos de swap estos son muy útiles si se necesita expandir el área de swap y no es posible asignar espacio en el disco duro para crear una partición mayor. La configuración de un archivo de swap es casi idéntica a la de la partición. La principal diferencia es que se tiene que crear el archivo antes de ejecutar *mkswap* y *swapon*.

Para crear un archivo de swap se utiliza el comando *dd*. Los principales aspectos antes de crear el archivo son el nombre del archivo y su tamaño en bloques. Para Linux un bloque tiene 1,024 bytes de tamaño. Por ejemplo, si se quisiera crear un archivo de nombre */swap* se teclearía lo siguiente:

```
# dd if=/dev/zero of=/swap_archivo bs=1024 count=10240
# mkswap /swap_archivo 10240
```

donde *of=/swap_archivo* especifica el nombre del archivo, y *count =10240* bloques o 10 Megabytes.

Se debe decir a *mkswap* el tamaño del archivo. Antes de ejecutar *swapon* sobre el archivo, se necesita asegurar de que el archivo esté completamente escrito en el disco. Se usa el comando *sync* para lograrlo. Ahora si se puede activar el archivo con el comando *swapon*, como en el siguiente ejemplo:

```
# sync
# swapon /swap_archivo
```

Si se tiene que deshacer del archivo de intercambio, hay que asegurarse que no esté activo esto se usa el comando *swapoff*, como ejemplo:

```
# swapoff /swap_archivo
```

Entonces ya se puede borrar con toda seguridad el archivo.

2.8 RESPALDO Y RECUPERACIÓN DE DATOS

Diferentes tipos de problemas pueden originar la pérdida de datos: eliminación accidental de archivos, una falla de hardware, etc. Por lo que es importante el conocer las formas de respaldar o recuperar dicha información(16).

Respaldos completos o incrementables, son los que copian todos los archivos. Los comandos relativamente simples para la creación de respaldos son : **tar** y **cpio.**

Desde el punto de vista del administrador, el sistema de archivo debe respaldarse de acuerdo con algún proceso automatizado, de preferencia cuando el sistema no se encuentre en uso, y con la menor intervención posible de los operadores. Además, se debe tener un plan de respaldo que satisfaga las necesidades y que haga posible la restauración de copias recientes de archivos, utilizando una combinación de respaldos completos e incrementables. Un respaldo completo es como se dijo inicialmente, el que contiene todos los archivos del sistema. Y el respaldo incrementado es el que contiene archivos que han cambiado desde el último respaldo . Estos puede realizarse en diferentes niveles:

Nivel 0	Respaldo completo.
Nivel 1	Incrementado con respecto al último respaldo completo.
Nivel 2	Incrementado con respecto al último respaldo del nivel 1.

Realiza respaldos de datos con el comando `'tar'` siguiendo la siguiente secuencia:

Copia el directorio `/home` a la unidad de disco flexible `/dev/fd0` utilizando la siguiente sintaxis, no olvides instalar un disco vacío en la unidad de discos:

```
#tar cvf /dev/fd0 /home
```

Nota: la **c** indica la creación de un archivo, y la **f** especifica el archivo destino, en este caso es la unidad de disco y **v** *verbose*.

Se archiva nuevamente el directorio `/home`, solo que ahora se realiza en modo extendido **(v)**, comprimiéndolo **(z)** y creando un respaldo multivolumen **(M)**, para lo cual se teclea lo siguiente:

```
#tar  cvfzM /dev/fd0 /home | tee /root/indice
```

Nota: De ésta forma, un listado de los archivos copiados se dirige a **/root/indice.** Para continuar, se ejecuta de nueva cuenta el comando del inciso 1.a.

i. Crea una lista con el comando 'find' para visualizar si se efectuaron con éxito las modificaciones realizadas en el inciso b, marcando :

```
#find /home -type -f print > bkuplust; tar cvfzM /dev/fd0 'cat bkplst' | tree
homeindex
```

Se restaura **(x)** el archivo `/home/ftp/bin/cpio` desde la unidad de disco usando los siguientes comandos:

```
# rm -/home/ftp/bin/cpio  Borrar el archivo para restaurarlo posteriormente
# cd /
# tar  xvzf /dev/fd0 /home/ftp/bin/cpio
```

Si se logró restaurar el archivo `cpio` con éxito, entonces se borra todo el directorio `/home` y después se restaura. Si se logro restaurar el archivo `cpio` no se procede. Se repiten los incisos 1.a, 1.b y 1.c tanto como sea necesario.

```
# cd /
# ls -l    Apunta la fecha de creación del directorio  /home
# rm -rf  /home Borrar el sub-directorio
# tar xvzf  /dev/fd0 Restaurar el directorio borrado accidentalmente
# ls -l    Compara la fecha de creación actual con la anterior
```

Se crea el archivo `/root/backup.tar` que contenga un respaldo del directorio `/home` y todos sus directorios y subdirectorios.

```
# cd /root
# tar cvf backup.tar /home
# ls -l    Comprobar que se creo backup.tar, apunta el tamaño en bytes
```

Las ventajas de comprimir la información después de respaldarla son :

```
# gzip backup.tar
# ls -l    Se comprueba que se creó backup.tar.gz
```

Un método alterno es:

```
# tar cvzf backup.tgz /home
# ls -l    Se comprueba el tamaño en bytes de los archivos backup.tar, backup.taz y
backup.tar
```

El directorio llamado **práctica** contiene varios archivos. Entonces se crea un archivo **tar** en `/home/red` que incluya solo el contenido de práctica sin un registro para el directorio, tecleando lo siguiente:

```
#pwd <enter>
/home/red
#cd práctica <enter>
/home/red/práctica
#tar cvf ../práctica.tar * <enter>
```

Se crea otro archivo **tar** comenzando desde el directorio padre de **práctica** y se especifica el nombre del directorio donde se va a guardar, para ello se sigue la siguiente sintaxis:

```
#pwd <enter>
/home/red
#tar cvf práctica.tar practica <enter>
```

Para ver el contenido de un respaldo que ya se hizo, se usa la opción **(t)**:

```
# cd /root
# tar tvzf backup.tgz | more
# tar tvf /dev/fd0 | more
```

Se realizan respaldos de datos con el comando `'cpio'` siguiendo la siguiente secuencia:

Se copian los archivos del directorio `/home` al dispositivo `/dev/fd0` usando el comando `cpio`:

```
#ls | cpio -oc > /dev/fd0
```

Nota: El comando `cpio` toma la entrada estándar y la copia a la salida estándar. En este ejemplo el comando `ls` envía a la salida estándar los nombres de los archivos. `cpio` los toma con entrada y los copia a la salida estándar, solo que ésta se encuentra redireccionada al dispositivo `/dev/fd0`. `cpio` con la opción **-o** hace una copia de un archivo en la salida estándar.

La forma más común de usar `cpio` es con el comando `find`. Ambos forman la mancuerna infalible y son los comandos más portables entre sistemas UNIX.

Se copian todo el directorio `/home` a un archivo llamado `/tmp/res.cpio`

```
# find /home -depth -print | cpio -ocv -O/tmp/res.cpio
```

Find procesará todos los archivos y después los directorios **(-depth)** y los enviará a la salida estándar **(-print)**. El comando cpio tomará los nombres de los archivos, los copiará **(-o)** y creará los directorios necesarios **(-c)** e indicará su estado **(-v)**. La salida de cpio se puede redireccionar como en el inciso 2.a o se pude usar la opción `-O`. El archivo de salida tendrá formato de cpio.

Para ver el contenido del archivo de respaldo con formato de cpio. Se ejecuta el siguiente comando:

```
# cpio -icvt -I/tmp/res.cpio
```

o se pudo haber ejecutado:

```
# cpio -icvt < /tmp/res.cpio
```

d Extrae **(i)** los archivos del dispositivo `/dev/fd0` y crea un índice en el archivo bkup.indx:

```
#cpio -it < /dev/fd0 > bkup.indx
```

Nota: La **t** indica que se crea una tabla de contenido de la entrada

Borra el archivo `/home/ftp/bin/gzip`

```
# rm /home/ftp/bin/gzip
rm: remove `/home/ftp/bin/gzip'? y
```

Restaura el archivo anterior, el cual fue borrado accidentalmente.

```
# cpio -icvd -I/dev/fd0
```

Comprueba si el archivo fue restaurado.

```
# ls -l /home/ftp/bin/gzip
```

Ahora se respaldan los archivos que se modificaron del subdirectorio `/home`. Utilizando el comando `'find'`, se saca una lista de todos los archivos `/home` que se modificaron y con `cpio` se hace el respaldo.

```
#find /home -mtime 1 -type f -print | cpio -ocv -O/dev/fd0
```

Una forma muy segura de copiar toda una estructura de directorios, preservando los permisos, dueños, grupos, ligas y fechas de creación es la siguiente:

```
# mkdir /users/newhome    Crear el directorio destino.
# cd /home        Colocarse en el directorio por copiar.
# find . -depth -print | cpio -pdmv /users/newhome
```

Copiar los archivos.

`# rm -rf /home` Opcional. Borrar el directorio origen, sólo si se desea mover el directorio.

2.9 RPM (RedHat Package Manager)

Una de las grandes tareas de la administración de sistemas es la instalación, actualización y borrado del software, ya que se necesita conocer donde se encuentran los archivos, programas, paginas del manual y otras documentaciones(5).

Ante esta situación Linux tiene un programa que facilita las tareas anteriores y se llama *RPM* (Redhat Package Manager ó manejador de paquetes de RedHat). En este programa se usan paquetes con formato RPM. Un paquete RPM consiste de un conjunto de archivos y describe la información del nombre, versión y contenido. Los paquetes creados por una función particular de Linux de este comando.

RPM tiene 11 modos de operación. Donde las primeras 5 son para la administración de los paquetes:

Instalar. Instala un paquete con las opciones obtenidas y la sintaxis es la siguiente:

rpm -i [opciones] [paquetes]

Actualizar. Instala la actualización del paquete con las opciones obtenidas usando la siguiente sintaxis:

rpm -U [opciones] [paquetes]

Borrar. Desinstala o borra el paquete con las opciones obtenidas (borra todas las posibles dependencias). Este modo tiene la siguiente sintaxis:

rpm -e [opciones] [paquete]

Buscar. Busca si el paquete está instalado y dónde está. Tiene la siguiente sintaxis:

rpm -q [opciones]

Verificar. Compara el paquete instalado con el original. La comparación incluye tamaño, un checksum, permisos, tipo, dueño y grupo de cada archivo. El modo de usarlo es:

rpm -V | --verify [opciones]

Los otros 6 modos son de distribución de software para crear o modificar:

Construir. Crea el paquete RPM . Este es muy usado por los desarrolladores de software, la sintaxis es:

rpm -b0 [opciones] paquetes

Reconstruir la base de datos. Reconstruye la base de datos con la configuración de la información del paquete. Se usa la siguiente sintaxis de este modo:

rpm --rebuilddb

Verifica firma. Revisa este paquete e integra si el original es correcto. Este modo de verificar la firma digital de un paquete hecho.

Pone dueño y grupo. Resetea el dueño y grupo original de los archivos pertenecientes de un paquete.

Fija permisos. Resetea los permisos de los archivos originales a los pertenecientes de un paquete.

Muestra RC. Muestra los valores del archivo rpmrc. El archivo rpmrc es usado para poner varios parámetros usados por RPM.

La tabla de RPM de los parámetros generales pueden usarse en algún modo:

Parámetro	Función
-w	*Imprime la información debug .*
--keep-temps	*No borra archivos temporales (/tmp/rpm-).*
--quiet	*Imprime tan pequeño como es posible, normalmente, despliega los mensajes de error.*
--help	*Imprime la ayuda.*
--version	*Imprime el número de versión de RPM.*
--rcfile <archivo>	*Especifica una posición de archivo diferente, estos como $HOME/rpmrc ó /etc/rpmrc*
--root	*Usa el directorio <dir> como el directorio de nivel de top de todas las operaciones.*

2.9.1 Instalación y actualización

Estas son algunas opciones

Parámetro	*Función*
--force	*Fuerza el reemplazo de un paquete o archivo.*
-h, --hash	*Imprime una marca "#", cuando un paquete se está instalando.*
--oldpackage	*Reemplaza un nuevo paquete con uno viejo. Normalmente RPM impide si se pregunta si escribe sobre un paquete nuevo.*
--percent	*Imprime el porcentaje de terminación durante la instalación.*
--replacefile	*Fuerza previamente archivos instalados desde otros paquetes para ser reemplazados.*
--replacepkgs	*Fuerza previamente paquetes instalados a ser reemplazados.*
--nodeps	*Salta los paquetes dependientes verificados antes de la instalación de paquete.*
--noscripts	*Salta los scripts pre y post instalación.*
--excludedocs	*Salta la instalación de documentación.*
--includedocs	*Instala la documentación de archivos.*
--test	*Una preforma de instalación pero sin tal acción. Esta función es buena porque verifica si hay problemas.*
-U --upgrade	*Instala el nuevo paquete sobre uno viejo. Remueve el viejo paquete.*

Ahora se instala el paquete openmotif, para esto necesitamos el paquete donde se puede encontrarlo vía ftp en ftp://rha.fi-b.unam.mx/pub/rpm/.

```
#rpm -ivh openmotif-2.1.30-4_MLI.i386.rpm
```

Se actualiza el paquete motif

```
#rpm -Uvh -force openmotif-2.1.30-4_MLI.i386.rpm
```

2.9.2 Borrado

La opción de borrado es una variante de la opción de instalación con respecto a las opciones, ya no muestra el porcentaje de desinstalación, pero busca que archivos están instalados.

Parámetro	Función
--noscripts	*Salta los scripts de pre y post desinstalación.*
--nodeps	*Salta el chequeo dependientes antes de la desinstalación*
--test	*Ejecuta los pasos de desinstalación sin borrar algo.*

Para desinstalar el programa openmotif se teclea:

```
#rpm -e openmotif-2.1.30-4_MLI
```

2.9.3 Query

Usando la opción "query" se determina qué paquetes se tienen instalados en el sistema, cuántos paquetes e información de cada uno de ellos ya que si se tiene uno no deseado se desinstala.

Parámetro	***Función***
-a	*Busca todos los paquetes instalados.*
--whatrequieres capabilty	*Paquetes colocados, stos requieren una capacidad particular.*
--whatprovides virtual	*Paquetes colocados, éstos requieren una capacidad virtual.*
-f <archivo>	*Localiza el paquete del cual proviene el archivo preguntado.*
-i	*Muestra información del paquete.*
--provides	*Muestra de qué paquete proviene.*
-l	*Lista los archivos del paquete.*
-s	*Muestra el estado de los archivos en el paquete (implica -l), el estado de cada archivo en termino si es normal, no instalado o reemplazado.*
-d	*Lista únicamente los archivos de documentación.*
-c	*Lista únicamente los archivos de configuración.*
--scripts	*Lista los shells scripts para la desinstalación e instalación.*
--dump	*Lista los archivos con la siguiente información:* ***path size mtime md5sum mode owner group lsconfig isdoc rdev symlink****. Estos parámetros deben ser usados con las banderas de* ***-l, -c, -d.***

Se verán todos los paquetes instalados:

```
#rpm -q -a
```

También los archivos de configuración del producto samba.

```
#rpm -q -c samba
```

No se conoce de que paquete viene el archivo */etc/*securetty y se quiere saber:

```
#rpm -qf  /etc/securetty
```

En unos ejemplos anteriores vimos como mostrar los archivos de configuración de SAMBA, pero se necesita un poco mas de información de los paquetes se teclea:

```
#rpm -qi samba
```

2.9.4 Verificar

Esta opción verifica los archivos que por alguna razón fueron alterados, este tipo de verificación es con respecto al tamaño, MD5sum (checksum), permisos, dueño, grupo, etc. Si se modifica el archivo */etc/exports* se puede saber si se corremos la siguiente opción:

```
#rpm -V nfs-server
```

Veremos algo similar:

```
S.5....T  c    /etc/exports
```

Donde la parte importante es la "c" que indica cambio y los otros caracteres cual fue el cambio, en este caso fue el tamaño, MD5 y la fecha respectivamente en el */etc/exports*. Ahora para ver más detalles se usará el archivo */etc/aliases*:

```
#ls -l  /etc/aliases
#rpm -qf  /etc/aliases
#rpm -V sendmail-8.x.y-z
```

Posteriormente se cambia al dueño "nobody" y se compara con el comando rpm:

```
#chown nobody /etc/aliases

#rpm -V sendmail-8.x.y-z
```

Al ver lo que sucede se cambia de dueño al archivo */etc/aliases* con los datos que se escribieron anteriormente. Ahora se observa lo qué significan los caracteres de los archivos cuando se usa *rpm*:

Carácter	*Atributo*	*Descripción*
5	*MD5 sum*	*Hace un checksum al archivo.*
S	*Tamaño del archivo*	*Cambio del archivo en bytes.*
L	*Liga*	*Liga simbólica.*
T	*Mtime*	*Modificación de fecha al archivo.*
D	*Dispositivo*	*Especificando que es un archivo de dispositivo.*
U	*Usuario*	*El archivo cambió de dueño.*
G	*Grupo*	*El archivo cambió de grupo.*
M	*Modo*	*Los permisos fueron modificados.*

Estos son los 8 caracteres que aparecen antes que la "c" (de cambio) y del "archivo" (Nótese que archivo se denomina como archivo regular, directorio, liga, etc.), pero si en vez

que aparezcan estos caracteres se muestra un punto (.) significan que se ha alterado este atributo. Para ver si todos los archivos fueron modificados se puede teclear:

`#rpm -Va > alterados` y posteriormente el resultado se encontrará dentro del archivo *alterados.*

2.10 CONFIGURACIÓN DE NFS

NFS (Networking File System o Sistemas de Archivo en Red) fue desarrollado y llevado a la mercadotecnia por Sun MicroSystems Inc. en 1984. Sun tiene la filosofía de distribución y sistemas abiertos. Así, desde el comienzo NFS fue diseñado para permitir la conexión de computadoras de diferentes sistemas operativos corriendo en ellas en redes homogéneas(11). Sin embargo, la realidad de NFS es que pudo ser usado en redes heterogéneas y fue probablemente el gran avance sobre otros productos como el Remote File System de AT&T (RFS) o el Andrew File System (AFS).

El NFS es el servicio más complejo de los que ofrecen RPC. Permite acceder a los archivos remotos exactamente igual que si fueran locales. NFS ofrece numerosas ventajas:

Los datos accedidos por todo tipo de usuarios pueden mantenerse en un nodo central, con clientes que montan los directorios en el momento de arrancar, esto es transparente a los usuarios. Si además se instala NIS, los usuarios podrían entrar y trabajar de forma transparente en cualquiera de las computadoras.

Los datos que consumen grandes cantidades de espacio en disco pueden mantenerse en un solo nodo.

Los datos de administración también pueden mantenerse en un solo nodo. No será necesario usar el rcp para instalar el mismo archivo.

Un aspecto de la transparencia es la velocidad de acceso de datos sobre la red, la cual debe ser alta no es notificable a diferencia del acceso a disco local. La meta original del desarrollo de NFS es el 80% de los datos con respecto al disco local.

El NFS de Linux principalmente fue obra de Rick Sladkey apoyándose en programas de Mark Shand. Donde la implantación del cliente será en el nuevo VFS (Virtual File Systems) y no necesita del código *biod.* NFS esta basado en un protocolo acordado cual largo corresponde al acordado definido con el modelo OSI:

Aplicación		mount, NFS, yp, bind
Presentación		XDR
Sesión		RPC
Transporte		TCP, UDP
Red		IP
Liga		ETHERNET
Físico		

2.10.1 Llamada de procedimiento remoto

La llamada de procedimiento remoto (RPC) es la capa de sesión de protocolo. La ejecución de un RPC consiste en los siguiente pasos:

1) Activación por el programa cliente. Los parámetros de solicitud son empaquetados dentro de un paquete de datos.

2) Mandando solicitudes y desempaquetamiento de los parámetros en el programa del servidor.

3) Ejecución de la solicitud (el procedimiento) en el servidor.

4) Empaquetar y retornar de los resultados del cliente.

5) Desempaquetar los resultados por el cliente y continuación de los programas normales de ejecución.

2.10.2 Port mapper

El servicio portmapper opera semejante a un servicio, esencialmente controla una tabla de mapeo de relación de programas RPC and/or números de versiones locales de TCP ó UDP de números de puertos, como es un servicio y disponibilidad(6). El portmapper ocupa el puerto 111 en ambos TCP y UDP.

2.10.3 Protocolo de cabecera RPC

El protocolo RPC es el chasis de transporte solicitado al servidor, cuando el flujo está dentro del procedimiento. La programación puede definir un número arbitrario de procedimientos entre el cliente y el servidor, con arbitrarias opciones de parámetro y resultados. Un funcionamiento de grupo coherente de procedimiento es llamado un servicio de RPC. Cada servicio es asignado a un llamado del numero de programa.

2.10.4 Protocolo Portmapper

Cliente y servidor ambos se comunican con el portmapper vía RPC. Los procedimientos RPC se definen en el protocolo portmapper. No es apropiado obtener la descripción exacta de los procedimientos y el resultado de este punto.

2.10.5 Protocolo de NFS

El protocolo NFS consiste en un número de procedimientos RPC, se construye con el mismo principio del protocolo portmapper. La especificación del número de puerto de 2049. Cuando alguien accede a un archivo remoto, el núcleo manda una llamada RPC al programa *nfsd* (*daemon* de NFS) del nodo remoto. Esta llamada incluye el descriptor de archivo, el nombre del archivo a acceder y los identificadores de usuario y de grupo del demandante. Estos identificadores se usan para verificar los permisos de acceso en la computadora remota, con lo que los usuarios de ambas computadoras deberían ser los mismos.

En varias aplicaciones de UNIX, las funcionalidades del cliente y servidor NFS, se implantan como d*aemons* de nivel del núcleo o al iniciar el sistema operativo. Se trata del programa *biod* (Block I/O *Daemon*) en el cliente. En cuanto a Linux el programa *biod* no se utiliza porque este se implantó en el VFS.

2.10.6 Daemons NFS

Si se desea proporcionar un servicio NFS a otras computadoras, se deberá ejecutar en el servidor los programas *nfsd y mountd.* Son programas basados en RPC, por los que no son arrancados por el *daemons* inetd, sino lanzados como *daemons* de tiempo de arranque y registrados en el mapeador de puertos de RPC.

Para realizar esta comunicación necesitamos dos computadoras Linux para hacer la interpretación de cliente-servidor, por esto necesitaremos las direcciones IP correspondientes, si no se sabe se usa el comando *ifconfig*, se corre tanto en el cliente como en el servidor.

Para probar si el cliente y el servidor tenga el filesystem de NFS en el kernel, se ve a través del siguiente archivo:

```
#cat /proc/filesystems
minix
ext2
msdos
nodev proc
nodev nfs
```

Si no se tiene la palabra de nfs entonces se recompila el kernel.

2.10.7 Configuración del servidor

Aquí se harán modificaciones en un archivo y se manipulan dos archivos ejecutables. Para poder compartir directorios en red, se necesita decir a los *daemons* de NFS cuáles se quieren y estos *daemons* leen el archivo */etc/exports*.

El archivo *exports* tiene el siguiente aspecto:

```
/directorio    nombre_de_la_computadora_ó_dir._IP(opciones)
```

```
Ejemplo:
#cat /etc/exports
/home      computadora1(rw) 192.168.2.10(rw) 192.168.2.54(rw)
/tmp       192.168.2.150(ro)
/usr/tmp   (ro)
```

Cada línea especifica un directorio, lista de la(s) computadora(s) que pueden acceder ella por medio de NFS. En la lista de la computadora puede aparecer con el nombre completo de la computadora o el numero de Internet completo. Aunque se pueden usar los comodines (* y ?) que tienen el shell. Si no lleva el nombre de la computadora (como en el caso de /usr/tmp) entonces se exportará a todas las computadoras de Internet(11).

Atrás del nombre de la computadora autorizada, se encerrará entre paréntesis el conjunto de opciones separadas por comas. Dichas opciones son:

insecure	Permitir acceso no autentificado desde ese nodo.
unix-rpc	Requiere autentificación RPC del dominio UNIX para este nodo. Se trata simplemente de que las peticiones se originen en un puerto reservado (es decir inferior a 1024). Esta opción está activa por defecto.
kerberos	Requiere autentificación Kerberos. Tampoco se ha implementado aun.
root_squash	Se trata de una opción de seguridad que niega acceso a archivos del superusuario a las peticiones con uid 0 o 65535 (este último se le asocia al usuario nobody).
no_root_squash	Evita la restricción anterior.
ro	Monta la jerarquía de archivos solo lectura.
rw	Monta la jerarquía con permisos para leer y escribir.

link_relative Convierte enlaces simbólicos absolutos (que comienzan con un barra de directorio, "/") en enlaces relativos colocados los prefijos ../ que sean necesarios para hacer que apunten a la raíz del servidor. Esta opción solo tiene sentido cuando se monta un sistema de archivos completo y no solo un directorio.

link_absolute Deja los enlaces absolutos como estaban.

map_identity Esta opción indica al servidor que el cliente utiliza el mismo mapa de uids y gids que el servidor.

map_daemon Esta opción indica al servidor NFS que no comparte el mapa de usuarios con el del cliente. Con ello, las comparaciones de uids y gids se harán mediante una lista de mapeado entre ambos que se construirá llamando al *daemon* ugidd del cliente.

Ahora se modifica ese archivo para que se tenga el siguiente formato:

```
#vi /etc/exports

/tmp            direccion_IP_cliente(rw,no_root_squash)
/mnt/floppy     direccion_IP_cliente(rw)
```

Se salva el archivo */etc/exports* y se requiere que los *daemons* tengan esta configuración en el archivo para lo cual se usarán los scripts:

```
#/etc/rc.d/init.d/nfs stop
#/etc/rc.d/init.d/nfs start
```

Si no se marca ningún error se ha acabado con la configuración del servidor y ahora se continuará con el cliente, pero antes se necesita un floppy para montarlo en el directorio */mnt/floppy* del servidor:

```
#mount /dev/fd0 /mnt/floopy
```

2.10.8 Configuración del cliente

Al estar en el cliente y se monta el directorio *tmp* que se encuentra en el servidor y se crea un directorio que sea la IP del servidor:

```
#mkdir /direccion_IP_del_servidor
#mkdir /direccion_IP_del_servidor/tmp
#mkdir /direccion_IP_del_servidor/floppy
#mount -t nfs nombre_servidor:/tmp  /direccion_IP_del_servidor/tmp
```

si no marca error el montaje se puede ver con mount:

```
#mount
```

Si se quiere que este directorio se monte cada vez que el cliente inicie se pueden poner las siguiente líneas en el archivo *etc/fstab*. Un ejemplo de este archivo es:

/dev/hda1	*/*	*ext2*	*defaults*	*1 1*
/dev/hda2	*swap*	*swap*	*defaults*	*0 0*
servidor:/tmp	***/direccion_IP_servidor***	***nfs***	***defaults***	***1 1***

Y con ello se logra cada vez que inicie la computadora cliente que el cliente busque ese directorio del servidor y cuando lo encuentre lo monta en su directorio, el comando *mount* tiene opciones para montar vía NFS que son las siguientes:

rsize=n y wsize=n	Especifican el tamaño del datagrama utilizado por el cliente NFS en las peticiones de lectura y escritura, respectivamente. Por defecto, cada una de ellas vale 1024 bytes, dados los limites del tamaño del datagrama UDP ya comentados.
timeo=n	Esta opción establece el tiempo máximo de espera de respuesta a una petición del cliente NFS; en centésimas de segundo. Por defecto, este valor es de 0.7 segundos.
hard	Marca el montaje del volumen como físico. Es un valor por defecto.
soft	Hace que el montaje sea solo lógico (opuesto al anterior).
intr	Esta opción habilita la posibilidad de que una señal interrumpa una espera por NFS. Es útil para poder abortarla cuando el servidor no responde.

Al montar el floopy que se encuentra en el servidor en el directorio */mnt/floppy* al cliente se tiene:

```
#mount servidor:/mnt/floppy /direccion_IP_del_servidor/floppy
```

Con ello se obtienen los datos de los directorios que se quiere del servidor. Para ver que computadoras ha montado nuestro directorio (el caso del servidor), se tiene un comando llamado:

```
#showmount -a
```

Con ello se tiene una forma de exportar y compartir archivos en una red bajo NFS.

2.11 CONFIGURACIÓN DEL PROGRAMA NIS (NETWORKING INFORMATION SERVICES)

Más y más computadoras LINUX son instaladas como parte de una red de computadoras. Para simplificar la administración en red, más redes corren el Servicio de Información en red (NIS). Las computadoras de LINUX pueden tener cierto avance de la existencia de un servicio NIS o estar provistas por el servidor NIS. Se puede también (con las librerías NYS) actuar como un cliente NIS+.

2.11.1 El portmapper RPC

Al ejecutar el programa mencionado se necesita correr el programa "/usr/sbin/rpc.portmap". Algunas distribuciones LINUX tienen el código en *las siguientes rutas: /etc/rc.d/inet2* (para Slackware) o */etc/rc.d/init.d/portmap* (para RedHat) e inicializan este *daemon.* El RPC portmaper es un servicio que convierte el programa RPC en programas dentro de protocolos TCP/IP (o UDP/IP) por números de puertos. Esta corriendo las llamadas RPC, de servidores RPC en computadoras clientes. Cuando un cliente desea hacer una llamada RPC obtiene un numero de programa, el primer socket portmap en el servidor determina el número de puerto por donde manda los paquetes RPC.

Con esto se puede uno dar cuenta que existe otro protocolo llamado UDP, y para saber un poco más sobre el y se darán los conceptos de dicho protocolo.

2.11.2 PROTOCOLOS DE DATAGRAMAS DE USUARIO (UDP)

Recordemos que los protocolos no orientados a conexión no proporcionan fiabilidad ni mecanismos de control de flujo. No proporcionan procedimientos de recuperación de errores. UDP es un protocolo no orientado a conexión. Se utiliza a veces como sustituto de TCP cuando no hay que utilizar los servicios de éste, por ejemplo, varios protocolos del nivel de aplicación, como el Protocolo de Transferencia de Datos Trivial (TFTP) y la Llamada de Procedimiento Remoto (RPC) utilizan UDP.

UDP sirve como interfaz de aplicación simple para IP; ya que incluye mecanismo de fiabilidad, control de flujo y sin medidas de recuperación de errores, sirve únicamente como multiplexor/demultiplexor de envío y recepción del trafico de IP.

UDP hace uso del concepto de puerto para dirigir los datagramas hacia las aplicaciones del nivel superior apropiadas. El datagrama de UDP contiene un numero de puerto de destino que es utilizado por el modulo de UDP para enviar el tráfico al receptor adecuado.

2.11.3 FORMATO DEL MENSAJE UDP

Quizá la mejor forma de explicar este protocolo sea examinar el mensaje y los campos que lo componen(9). Como muestra la figura el formato es demasiado simple que incluye los siguiente campos:

32 BITS	
Puerto de Fuente	Puerto de Destino
Longitud	Checksum
Datos	

PUERTO DE FUENTE: Este valor identifica el puerto del proceso de aplicación remitente. Este campo es opcional. Si no se utiliza, se pone a 0.

PUERTO DE DESTINO: Este valor identifica el proceso de recepción en la computadora destino.

LONGITUD: Este valor indica la longitud del datagrama del usuario, incluyendo la cabecera y los datos. La longitud mínima es de 8 bits.

CHECKSUM: Este valor contiene el valor del complemento a 1 en 16 bits de la suma de la subcabecera de IP, la cabecera de UDP y los datos. Se realiza también el checksum de los campos de relleno (si es necesario que el mensaje contenga un número de octetos que sea un múltiplo de 2).

Poco más se puede decir de UDP; representa el nivel de servicio mínimo que utilizan muchos sistemas de aplicación basados en transacciones; es sin embargo, muy útil en los casos en los que son necesarios los servicios de TCP.

Para saber qué servicios ofrecen TCP/IP y UDP, será necesario ver el archivo */etc/services* para saber qué programas usa cada protocolo, ó ambos.

También se necesitan diversos programas (ypbind, ypcat, yppoll, ypmatch), pero el más importante es el ypbind. Este programa deberá correr todo el tiempo (en el cliente). Se usará el servidor *con dirección IP 192.168.2.1 (para este tipo de pruebas)* con sus archivos ya configurados, donde el dominio de las páginas amarillas se denominó *Musica*. Pero surge una pregunta:

Como configurar un servidor de páginas amarillas (ypserv) :

Esto se hará en dos partes:

I)En nuestro servidor asignado se repetirá lo que se hizo en el servidor 192.168.2.1.

II)Después que hacer la configuración de tal servidor, se configurará la computadora asignada para que sea un cliente y lea las páginas amarillas del servidor 192.168.2.1.

2.11.4 CONFIGURACION DEL SERVIDOR

Se edita el archivo *Makefile* que se encuentra en /var/yp y se pone entre comentarios desde la palabra netid y se verá de la siguiente forma (aproximadamente se encuentra en la línea 91):

all : passwd hosts group netid networks protocols rpc services netgrp \ mail shadow ypservers publickey ethers # amd.home bootparams:

Como queda :
all : passwd hosts group #netid networks protocols rpc services netgrp \ mail shadow ypservers publickey ethers # amd.home bootparams.

Se dará de baja en el servidor el *daemon* de las paginas amarillas "ypserv" con la siguiente sintaxis:

```
#/etc/rc.d/init.d/ypserv.init stop
```

veremos qué dominio tiene el servidor:

```
#domainname
(none)
#    #domainname curso
#    #domainname
curso
#
```

el cual nos dice el dominio asignado, ahora se crean las páginas amarillas con el comando *make* en el directorio /var/yp:

```
# cd /var/yp
# make
.....................
#
```

Después de crearlas, se hace un directorio con ese nombre, ahora se levanta el *daemon* ypserv.init, pero antes se cambia el dominio y que sea del servidor llamado 192.168.2.1, (dominio Musica):

```
#domainname Musica
#
#/etc/rc.d/init.d/ypserv.init start
```

los programas que se ve que están funcionando son los archivos ypasswdd, portmapper y ypserv, también se pueden usar la siguiente expresión para ver si está corriendo el *daemon* ypserver o unos de los antes mencionados:

```
#rpcinfo -u localhost ypserver
100004 version 2 ready and waiting
#
```

lo que indica que el programa está listo y en espera de datos.

2.11.5 CONFIGURACION DEL CLIENTE

Ahora se configurará el cliente de tal manera que lea las páginas amarillas del servidor. Primero se creará un archivo para que sepa de qué computadora esta exportando las "yellow pages":

```
#vi /etc/yp.conf
ypserver 192.168.2.1
:wq
```

Y posteriormente se levantará el *daemon* ypbind con la siguiente sintaxis:

```
#/usr/sbin/ypbind
#
```

Para ver si esta levantado usaremos la siguiente expresión:

```
#rpcinfo  -p
program     vers    proto    port
100000       2      tcp       111  portmapper
100000       2      udp       111  portmapper
100005       1      udp       777  mountd
100005       1      tcp       779  mountd
100003       2      udp      2049  nfs
100003       2      tcp      2049  nfs
100004       2      udp       815  ypserv
100004       1      udp       815  ypserv
100004       2      tcp       818  ypserv
100007       2      udp       637  ypbind
100007       2      tcp       639  ypbind
100009       1      udp       824  yppasswdd
300019       1      udp       836
#
```

Para ver si se exportan las tales páginas amarillas, (las del servidor 192.168.2.1) se usará el comando *ypcat*, la sintaxis siguiente (el comando ve las páginas amarillas y las transforma para verlas en la terminal) :

```
#ypcat passwd  |  more
```

o la siguiente expresión:

```
#ypcat group | more
```

ahí se verán los archivos */etc/passwd, /etc/group y /etc/hosts* del servidor 192.168.2.1, ahora se creará un directorio en la raíz:

```
#cd /
#mkdir users
```

Se monta el directorio users del servidor 192.168.2.1:

```
#mount 192.168.2.1:/pub/users /users
```

En la computadora cliente se configuran los archivos */etc/passwd y /etc/group*, al final de estos se debe poner un signo mas (+):

Ejemplo:
++:*:0:0:::

ya que hicimos la configuración de los archivos */etc/passwd y /etc/group* de 192.168.2.1, usaremos un login del tal servidor:

```
login es NIS
passwd es Redes97
```

con ello se puede usar el espacio en disco de ese usuario y dar por terminada la configuración del NIS, pero hay una pregunta :

¿Se podrá cambiar el password desde una terminal usando NIS ?

La respuesta es no, pero con otro programa llamado *yppasswd*, sí se puede, a continuación se dará una explicación y cómo configurar este programa para poder cambiar el password.

En la distribución de los sistemas Linux, el passwd, el chsh y chfn solamente se pueden cambiar si el usuario esta dado de alta en la computadora local, pero como está leyendo las paginas amarillas del servidor no sirven entonces esos comandos, para ello se tienen el yppasswd, ypchfn y ypchsh, estos 2 últimos comandos pueden ser opciones del primero con la bandera -f y -l, respectivamente.

El yppasswd cambia el passwd del usuario que está en el NIS. El usuario tendrá que escribir su passwd con un mínimo de 6 caracteres.

El ypchsh o yppasswd -l cambia el shell del usuario NIS, y aparecerá el shell que se usa por default:

Login shell [/bin/bash]:

Si se quiere ese shell, solamente se dará ENTER, si no se deberá dar el shell que se quiere, pero siempre con la ruta absoluta donde se encuentra el shell.

El ypchfn ó yppasswd -f Cambia los datos que tiene registrados el usuario NIS, lo que se llama el GECOS, donde aparecerán los siguientes datos por cambiar:

Name [NIS]:
Location [C.U.]:
Office Phone []:
Home Phone []:

Si se quiere tener los mismos datos por default, solamente se teclea ENTER, si no se hace las modificaciones pertinentes.

La instalación del yppasswd es muy sencilla, se utilizan los siguientes comandos:

```
#cd /usr/bin
#mv passwd passwd.old
#chmod go-rwx passwd.old
#ln yppasswd passwd
```

Esta secuencia de sintaxis es:

- Cambiarse a /usr/bin
- Mover el comando passwd
- Quitar permisos para ejecutarlo, tanto al grupo como a otras personas
- Ligar el programa passwd con yppasswd, para poder así cambiar el passwd de la configuración NIS. Donde cabe señalar que el comando passwd.old solamente lo puede ejecutar el administrador . Esto se hace en las computadoras con clientes NIS.

2.12 MODIFICACIONES AL KERNEL

Los códigos del Kernel con nuevas características de Linux están disponibles en CD, sitios FTP, grupos de usuarios, etc.

La mayoría de las versiones del Kernel están numeradas con una versión y el nivel del parche, si tecleamos el comando:

#uname -r - Nos permite ver la versión del Kernel.
2.0.18
#

Donde el "2" es la característica mayor, "0" es la característica menor de la versión del Kernel, y "18" es el número de parche.

Los parches algunas veces están numerados en forma diferente, y no se requiere el código entero del Kernel para instalarlo. Solo se requiere el código del parche. En la mayoría de los casos, el parche se sobrepone en la sección del Kernel ya existente del código-fuente y una simple recompilación es todo lo que se necesita para instalarlo, en su mayoría los parches son modificaciones pequeñas del Kernel.

Se debe tener una versión del compilador gcc o g++ (compiladores GNU) actualizada, porque ocasionalmente las nuevas características del Kernel no serán soportadas por versiones antiguas de gcc o g++.

2.12.1 Modificación del kernel

Tecleamos:

 # cd /usr/src - En este directorio se encuentra el código fuente del Kernel.
 # ls -l - Se verifica la existencia del directorio Linux-2.0.18
 # cd Linux-2.0.18 - Se ubica en el directorio donde se encuentra el código fuente y es donde vamos a trabajar.

Hay que verificar que el archivo `/usr/scr/Linux-2.0.18/Makefile` contenga la línea ROOT_DEV = CURRENT. El cual es el dispositivo que es usado por la raíz del sistema de archivos cuando booteamos Linux.

Tecleamos:

 # vi Makefile - Con el editor vi se verifica el archivo.

a. Con la tecla flecha se mueve a través del archivo.
b. Localizar la linea ROOT_DEV=CURRENT

```
Esc
:q!                    - Se sale del archivo sin hacer modificación alguna.
#
```

El proceso de compilación empieza cuando se teclea el comando.

 # make config
- Aparecerán una gran cantidad de modificaciones que se puede hacer al Kernel.

También tenemos otras opciones:

#make menuconfig
-Aparece las mismas opciones pero con un menú mas amigable.

#make xconfig
-Si tenemos sesión gráfica.

Como por ejemplo:

Las respuestas se darán con "y" (sí) ó "n" de (no). Otros dispositivos típicos tienen opción de "m" (módulo), ya que estos no se pondrán dentro del kernel, pero como serán un módulo cargable. También si tenemos dudas esta la opción "?".

Estas opciones son para agregar, remover drivers, protocolos, filesystems, etc. se verán menús principales:

- Emulador matemático en el kernel.
- Discos y CDROM IDE soportados.
- Soporte para RED (TCP/IP, PPP, SLIP, Firewall)
- System V IPC
- Tipo de Procesador (386, 486, Pentium, Ppro)
- Soporte para SCSI (CDROM, Discos, Cintas, etc.)
- Dispositivo de red soportado (Tarjeta de red, PLIP)
- Sistema de archivos (minix, etxfs, etx2fs, msdos, umsdos, proc, iso9660, hpfs, nfs, etc.)
- Dispositivos tipo caracter (impresoras, busmouse, PS/2 mouse, etc.)
- Tarjeta de sonido
- Otras configuraciones

```
#
# Using defaults found in arch/i386/defconfig
#
*
* Code maturity level options
*
```

```
Prompt for development and/or incomplete code/drivers
(CONFIG_EXPERIMENTAL) [N/y/?]
*
* Loadable module support
*
Enable loadable module support (CONFIG_MODULES) [Y/n/?]
Set version information on all symbols for modules
(CONFIG_MODVERSIONS) [Y/n/?]
Kernel daemon support (e.g. autoload of modules)
(CONFIG_KERNELD) [Y/n/?] ?
```

Se deben que encontrar tres líneas que son las importantes para desactivar el puerto paralelo lp1. Se teclea una "n" para desactivar esa opción.

Nota: Si se tecleas ? aparecerá en la pantalla información de ayuda referente a esa opción. Para moverse con rapidez para localizar estas opciones basta con teclear enter ( ) después de cada petición.

```
PLIP (parallel port) support (CONFIG_PLIP) [M/n/y/?] N
IOMEGA Parallel Port ZIP drive SCSI support
(CONFIG_SCSI_PPA) [M/n/y/?] N
Parallel printer support (CONFIG_PRINTER) [M/n/y/?] N
```

Con esto se finaliza la configuración del kernel.

Limpieza y dependencias.

Se teclean los comandos:

```
# make dep ; make clean
```

'make dep' - Asegura que todas las dependencias, tales como los archivos include, estén en su lugar.

'make clean' - Este comando remueve todos los archivos objeto y algunas otras cosas de la versión anterior.

2.12.2 Compilación del kernel.

Se teclea el comando:

 # make zImage - Crea una imagen compactada del nuevo Kernel.

Se pueden crear los kernel: zImage, zlilo, bzImage, zdisk. Están comprimidos y se descomprimirán automáticamente cuando se ejecuten.

Ésta es la parte más tardada del proceso y depende de la computadora que se ocupe. En una computadora Pentium a 66 MHz toma aproximadamente 40 minutos y una Pentium a 233 Mhz con 32 Mb sé lleva aproximadamente 9 minutos.

2.12.3 Instalación del nuevo kernel

 # pwd
/usr/src/Linux
 # cd arch/i386/boot - Se mueve al directorio boot.
 # cp zImage /boot/zImage - Copia el nuevo kernel al directorio de arranque.
 # cd /etc - Se cambia de directorio para modificar el archivo de arranque LILO (Linux Loader).

 # vi lilo.conf - Edita el archivo lilo.conf con el editor vi.
a. Tecleamos **i** - Permite modificar el archivo.
El archivo es el siguiente.

```
map=/boot/map
install=/boot/boot.b
prompt
timeout= 50
image=/boot/vmlinuz
        label=Linux
        root=/dev/hda2
        read_only
```

b. Agregar las siguientes lineas al final del archivo.

```
image=/boot/zImage
label=Linux2
root=/dev/hda2
read_only
```

El archivo `lilo.conf` nos queda de la siguiente forma:

```
map=/boot/map
install=/boot/boot.b
prompt
timeout= 50
image=/boot/vmlinuz
        label=Linux
        root=/dev/hda2
        read_only
image=/boot/zImage
        label=Linux2
        root=/dev/hda2
        read_only
```

c. Tecleamos:

Esc

:wq - Sale del editor vi guardando los cambios hechos al archivo.

 # lilo - Actualiza el archivo lilo para tener una nueva forma de arranque.

Con estas modificaciones se tiene dos kernels para arrancar el sistema Linux. El anterior con el puerto paralelo lp1 activado (Linux) y el nuevo con el puerto paralelo lp1 desactivado (Linux2).

2.12.4 Verificación del kernel

Salir del sistema con el comando:

 # shutdown -h 0 - Salida de forma total de Linux.

Apagar la computadora y volverla a encender.

Cuando aparezca

LILO

Se teclea:

TAB ( ) - Aparecerán los nombres de los dos kernels con los que se puede activar el sistema Linux.

LILO: Linux2 - Arrancará con el nuevo Kernel.

Con el comando **'dmesg'** se verifica que no aparezca el puerto lp1.

 # dmesg | more - Permite ver el despliegue del comando por partes.

Se verifica que no se encuentre la linea siguiente:

lp1 at 0x0378, (polling)

Si no se encuentra hay que revisar que checar qué punto salió mal, y en el peor de los casos volver a empezar.

Volver a repetir el punto 4, pero en vez de cargar **'Linux2'** arrancar con el kernel anterior **'linux'** y se teclea el comando **'dmesg'** para ver que se encuentre la linea **lp1 at 0x0378, (polling)**.

2.12.5 Módulos

Al cargar los módulos en el kernel se puede "salvar" memoria y una fácil configuración. Estos módulos incluyen filesystems, drivers para tarjeta ethernet, drivers para impresora y más. Hay varios programas que se tienen para manejar los módulos como:

insmod, rmmod, ksyms, lsmod, genksyms, modprobe y depmod.

El comando *insmod* inserta un modulo dentro del kernel corriendo. Estos módulos tienen extensión o, por ejemplo:

```
#insmod drv_hola.o
```

Con ello se carga el driver, además si se quiere ver los módulos cargados será con el comando *lsmod*:

```
#lsmod
Module:    #pages:    Used by:
drv_hola        1
```

Ahora si se quiere remover él modulo, se usa el comando *rmmod*:

```
#rmmod drv_hola
```

Después de configurar el kernel, ya sea cuando se reinicia el sistema o acabando la opción de "*make config*" (o similares) dentro del directorio */usr/src/Linux*, teclear:

```
#make modules          -Preparar los módulos que se ejecutarán
```

Después de compilar todos los módulos estos se encontrarán en el archivo /usr/src/Linux/modules, los módulos que se ejecutarán. Ahora para que estos se ejecuten cada vez que se reinicie la computadora se ejecutará

```
#make modules_install
```

Lo cual instalará los módulos en el directorio /lib/modules/x.y.z donde x.y.z es la versión del kernel.

CAPITULO 3

3.1 MANEJO DE TERMINALES

Los dispositivos correspondientes a los cuatro puertos seriales son:

```
/dev/cua0, /dev/ttyS0   (COM1)   dirección 0x3f8   IRQ 4
/dev/cua1, /dev/ttyS1   (COM2)   dirección 0x2f8   IRQ 3
/dev/cua2, /dev/ttyS2   (COM3)   dirección 0x3e8   IRQ 4
/dev/cua3, /dev/ttyS3   (COM4)   dirección 0x2e8   IRQ 3
```

Los números mayor y menor de los dispositivos seriales en `/dev` son:

```
/dev/ttyS0 mayor 4, menor 64     /dev/cua0 mayor 5, menor 64
/dev/ttyS1 mayor 4, menor 65     /dev/cua1 mayor 5, menor 65
/dev/ttyS2 mayor 4, menor 66     /dev/cua2 mayor 5, menor 66
/dev/ttyS3 mayor 4, menor 67     /dev/cua3 mayor 5, menor 67
```

`'getty'` es un programa que maneja el proceso de login cuando se accede a una caja de Unix. Hay tres versiones que son comúnmente usadas en Linux: `'agetty'`, `'getty_ps'` y `'mgetty'`.

`'getty_ps'` es una versión de `'getty'` y fue escrita por Paul Sutcliffe Jr., Kris Gleason. El paquete de `'getty_ps'` contiene dos getties. El `'getty'` es usado para las consolas y dispositivos de terminal - y `'uugetty'` es usado para módems.

`'mgetty'` es una versión de `'getty'` escrita por Gert Doering. Además de manejar el proceso de login, `'mgetty'` provee soporte para fax a través de `'sendfax'`, el cual acompaña al `'mgetty'`. mgetty+sendfax 0.22 son la ultima versión de este paquete.

`'agetty'` es la tercer variación de `'getty'`. Esta fue escrita por W.Z. Venema. Este es una simple implantación de `'getty'`.

Requerimientos de Hardware

Dependiendo del tipo de puerto serial se necesitan conectores DB25 o DB9 hembra, la forma de configurar los conectores es la siguiente:

Para dos conectores DB25:

TxD	Transmite Dato	2 - 3	RxD	Recibe Dato
RxD	Recibe Dato	3 - 2	TxD	Transmite Dato
SG	Señal de Tierra	7 - 7	SG	Señal de Tierra

Para dos conectores DB9:

RxD	Recibe Dato	2 - 3	TxD	Transmite Dato
TxD	Transmite Dato	3 - 2	RxD	Recibe Dato
SG	Señal de Tierra	5 - 5	SG	Señal de Tierra

Para un conector DB9 y un conector DB25:

RxD	Recibe Dato	2 - 2	TxD	Transmite Dato
TxD	Transmite Dato	3 - 3	RxD	Recibe Dato
SG	Señal de Tierra	5 - 7	SG	Señal de Tierra

Como cada computadora tiene dos puertos seriales se pueden dar las siguientes combinaciones de conexión:

Servidor	Terminal
COM1	COM1
COM1	COM2
COM2	COM1
COM2	COM2

NOTA: Este desarrollo fue realizado tomando en cuenta la primera combinación COM1 - COM1, si se utiliza una combinación de puertos diferente se deben realizar los cambios pertinentes.

3.1.1 Configuración de una terminal tonta en Linux.

1. Arrancar el servidor de Linux.

2. En la computadora que va a quedar como terminal se carga el programa que emula una PC como terminal tonta. Tener copiado el programa en un disco 3.5" de HD. Este programa se encuentra en el servidor rha (132.248.59.5) vía ftp anónimo y en el siguiente directorio: /pub/serial

```
A:\>st240
```

a. Seleccionar la opción `1:Serial1` para aceptar la conexión a través del puerto serial 1,
o
b. Seleccionar la opción `2:Serial2` para aceptar la conexión a través del puerto serial 2.

La computadora estará en espera de datos.

3. En la computadora que se encuentra como servidor, se teclea el siguiente comando para comprobar que hay comunicación.

```
# cat /etc/issue > /dev/ttyS0
```

NOTA: Se entiende que el servidor se comunicara con la terminal a través del puerto serial 1 (COM1), que Linux lo conoce como `/dev/ttyS0`.

Si está bien configurado el cable y el programa emulador de terminal, está aceptando datos por el puerto que tiene conectado el cable, en la terminal debe aparecer un mensaje similar a este:

```
Red Hat Linux release 4.0 (Colgate)
kernel 2.0.18 on an i586
```

4. Para dar de alta una terminal en el servidor, se modifican los siguientes archivos.

a. `/etc/gettydefs` las modificaciones son las siguientes:

```
# 38400 bps Dumb Terminal entry
DT38400# B38400 CS8 CLOCAL # B38400 SANE -ISTRIP CLOCAL #@S @L login:
#DT3840
# 19200 bps Dumb Terminal entry
DT19200# B19200 CS8 CLOCAL # B19200 SANE -ISTRIP CLOCAL #@S @L login:
#DT1920
# 9600 bps Dumb Terminal entry
DT9600# B9600 CS8 CLOCAL # B9600 SANE -ISTRIP CLOCAL #@S @L login: #DT9600
```

Nota: Si no existen estas líneas habrá que capturarlas.

b. `/etc/inittab` las modificaciones son las siguientes:

```
S0:23456:respawn:/sbin/getty ttyS0 DT9600 vt100
```

5. Una vez que se han hecho modificaciones en el `/etc/inittab` es necesario que `init` lea nuevamente su archivo de configuración. Teclear el siguiente comando:

```
# init q
```

o bien:

```
# telinit q
```

6. Comprobar que el comando 'getty' esté corriendo. Debe encontrarse una línea similar. Se teclea el siguiente comando:

```
# ps -ax | grep ttyS0
519   S0   S   0:00   /sbin/getty   ttyS0   DT9600   vt100
```

Si el proceso ha sido creado por init, se obtendrá la salida anterior con lo que se concluye que el proceso de la terminal que se acaba de dar de alta se encuentra operando,

NOTA: El PID puede ser distinto.

7. La terminal se encuentra en espera del login y password del usuario, se teclea uno o dos enter para establecer la comunicación. En la pantalla debe aparecer algo similar a lo siguiente:

```
Red Hat Linux release 4.0 (Colgate)
kernel 2.0.18 on an i585

sol
```

8. Probar que la configuración y conexión es correcta entrando a sesión como usuario normal. Por ejemplo:

```
sol apollo
Password:

Last login: Tue Nov 26 18:13:17 on ttyS0
 apollo@sol apollo $
```

9. Para entrar como root agregar al archivo /etc/securetty la siguiente línea:

```
ttyS0
```

3.1.2 Configuración de un terminal en plataforma Windows

Después de haber logrado con éxito la conexión bajo terminales, ahora se hará con una tarjeta de red y bajo Windows 3.11. Ahora la siguiente cuestión es saber qué tarjeta tiene la computadora:

En esta sala hay 2 tipos:

```
Data Link 220

Data Link 250
```

Posteriormente se necesita un disco, además de la dirección IP asignada se tendrán que conectar al servidor 132.248.59.5 (o si ya tiene los drivers de las data link), haciendo un *ftp* anónimo, la sintaxis cuando se este dentro de la computadora será :

```
#ftp 132.248.59.5
Connected to 132.248.59.5
```

Luego pedirá el nombre del usuario:

```
Name (132.248.59.5 : mena):
```

El usuario que se aparece es el que se conecta, pero se usara el login *anonymous* (anónimo)

```
Name (132.248.59.5 : mena): anonymous
```

```
passwd:                                  # Login de una cuenta de su computadora
```

Ahora se pasa al directorio `pub/dos/win`

```
ftp> cd pub/dos/win
250 CWD command successful
ftp>
```

Ahora con un `ls` se verán los archivos que se encuentran

```
ftp> ls
200 PORT command successful.
150 Opening ASCII mode data connection for /bin/ls.
total 452
drwxrwxr-x  2 root   ftp      1024  Jul  3 17:04 .
drwxr-xr-x  4 root   ftp      1024  Jul  3 17:10 ..
-rw-rw-r--  1 root   ftp      4768  Jul  3 17:04  de220pd.com
-rw-rw-r--  1 root   ftp      4951  Jul  3 17:04  de220pd.sys
-rw-rw-r--  1 root   ftp      4768  Jul  3 17:04  de250pd.com
-rw-rw-r--  1 root   ftp      4951  Jul  3 17:04  de250pd.sys
-rw-rw-r--  1 root   ftp    144384  Jul  3 16:58  tcpman.exe
-rw-rw-r--  1 root   ftp     71168  Jul  3 16:58  trmptel.exe
-rw-rw-r--  1 root   ftp     65280  Jul  3 16:58  trumping.exe
-rw-rw-r--  1 root   ftp       935  Jul  3 16:58  trumpwsk.ini
-rw-rw-r--  1 root   ftp      3516  Jul  3 17:04  winpkt.com
-rw-rw-r--  1 root   ftp    159744  Jul  3 17:04  winsock.dll
226 Transfer complete.
ftp>
```

Ahora se traen los archivos

```
ftp> bin
200Type set to Y.
 # Transferencia en modo binario
ftp> hash
Hash mark printing on (1024 bytes/hash mark) señal para que
cada 1024 bytes imprima un "#"

ftp> prompt
Interactive mode off.          Modo de interacción desactivado
ftp> mget *

#########?????????????
```

Después de que la transferencia de archivos se ha completado, se sale con el comando exit

```
ftp>exit
221 Goodbye
```

#

Después, en la computadora cliente se crea un directorio llamado terminal

C:\> mkdir terminal
C:\>

Copiar los archivos a ese directorio :

C:\> cd terminal
C:\terminal> copy a:.* .*

Ahora se entra a Windows y se configura el tcpman.exe. En el administrador de programas, en la barra del menú en la opción archivo, se busca la opción ejecutar , luego se encuentra la opción examinar y en el directorio que se crea (llamado terminal) se busca el archivo tcpman.exe , después aceptar

Lo que se hará posteriormente será configurar este programa por ello se debe saber:

- La dirección IP de la computadora

- Dominio de la red

Cuando se abre el archivo se va al menú y se busca la opción file posteriormente la opción setup y ahí se hará la configuración, en la primera figura es una aproximación de lo que sale, y en la otra figura es como debe quedar.

Después de darle OK, pedirá que se cierre el programa para que se pueda configurar, pero además saldremos se saldrá de Windows para cargar los siguiente, paquetes:

C:\terminal>de2#0pd 0x69

Es el manejador de la tarjeta y el signo " # " es el tipo de tarjeta que tiene esa computadora..
C:\terminal>set ip=xxx.yyy.zzz.www

Es la dirección IP de la computadora en que se esta trabajando.

C:\terminal>winpkt 0x69

Es la conexión con *sockets* para Windows. Después se entra otra vez a Windows y se abre el archivo tcpman.exe, posteriormente se verá la configuración que se hizo antes de salir, y se abrirá el archivo trmptel, que se encuentra en el mismo lugar que tcpman, en los que aparecerá una ventana que dice :

Con ello se dará la dirección del servidor correspondiente y cuando haga la conexión con el servidor indicará el cómo conectarse al servidor en una terminal de Windows.

3.2 SAMBA

3.2.1 Introducción a SAMBA

SAMBA es una conjunto de programas el cual trabaja y aloja clientes accesando archivos e impresoras vía el protocolo SMB (Session Message Block). Inicialmente escrito para UNIX, ahora SAMBA también corre en Netware, AmigaDOS, OS/2, etc.

En la práctica, se puede redireccionar discos e impresoras UNIX y que sean clientes, tales como WFW3.11, Windows9.x, WindowsNT, Linux y OS/2. Esto es que un cliente genérico UNIX, suple parte del conjunto que aloja a usuarios y tiene una interfaz tipo ftp cuando se conecta una computadora UNIX a un servidor SAMBA, accesando a sus recursos vía el servicio smb. Con esto se obtienen la capacidad de que este servidor Linux trabaje mejor que un servidor Windows NT, únicamente teniendo funcionalidad y flexibilidad como está diseñado para hacerlo mas fácil a los administradores(5).

Los componentes de la suite son:

Smbd, el servidor smb. Este actúa con las conexiones desde clientes, dando todos los archivos, permisos y directorio de trabajo al usuario.

Nmbd es el servidor de nombre de NetBIOS, el cual ayuda a los clientes a localizar servidores, buscando y manejando dominios.

Smbclient, el programa cliente del host-UNIX.

Testprns un programa para hacer un test al servidor que accede a las impresoras.

Testparms, este programa hace un test a la configuración del archivo samba y verifica que esté correcto.

Smb.conf es el archivo de configuración SAMBA.

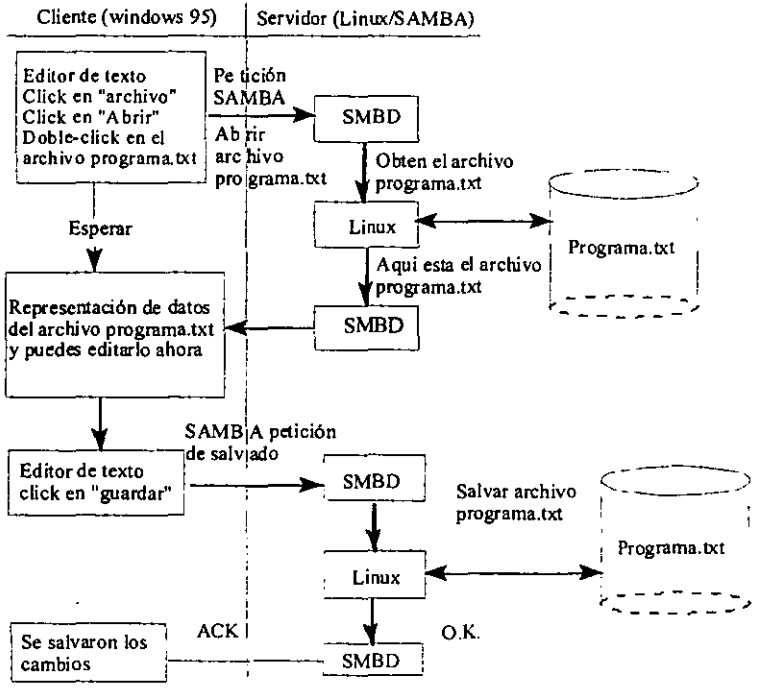
Smbstatus muestra lo compartido (recursos) exportado por SAMBA.

Smbfs, es un sistema de archivos que trabaja con el protocolo smb. Con esto se puede montar o compartir recursos con computadoras UNIX.

Smbmount es el comando que monta el smbfs de un sistema UNIX.

3.2.2 El protocolo SAMBA

El protocolo smb es responsable de la negociación de los archivos remotos y recursos como impresoras. Si se habla de un ejemplo más específico, usaremos un cliente Windows 95 y un servidor Linux, en este servidor tenemos un directorio mapeado a Windows 95 en el drive E. Este directorio tiene un archivo llamado *programa.txt*. El siguiente diagrama muestra el proceso que ocurre, cómo el cliente busca y accede al archivo desde el servidor.



3.2.3 Los *daemons* de SAMBA

El corazón de SAMBA son los programas smbd y nmbd, usualmente corren como *daemons*, todos como todo el tiempo. RedHat los corre por default.

El proceso nmbd habilita al servidor Linux a ser explorado por otras computadoras. El *daemon* smbd contiene los paquetes SMB y estos arriban a la red y negocia con el kernel de Linux a acceder estos recursos y compartirlos. Si un archivo es un recurso, una petición de impresión. La forma de exportar es:

Archivos	Impresoras
Linux <-->Linux	Linux<-->Linux
Windows <-->Linux	Windows <-->Linux

3.2.4 Archivo de configuración de SAMBA

El archivo llamado *smb.conf* tiene toda la configuración e información del uso smbd y nmbd. Smb.conf [Archivo]

```
# This is the main Samba configuration file. You should read the
# smb.conf(5) manual page in order to understand the options listed
# here. Samba has a huge number of configurable options (perhaps too
# many!) most of which are not shown in this example
#
# Any line which starts with a ; (semi-colon) or a # (hash)
# is a comment and is ignored. In this example we will use a
#
# for commentry and a ; for parts of the config file that you# may wish to
enable
#
# NOTE: Whenever you modify this file you should run the command "testparm"
# to check that you have not many any basic syntactic errors. #
#======================= Global Settings =====================================
[global]

# workgroup = NT-Domain-Name or Workgroup-Name  workgroup = WORKGROUP
# server string is the equivalent of the NT Description field
server string = Samba Server
# This option is important for security. It allows you to restrict
# connections to machines which are on your local network. The
# following example restricts access to two C class networks and
# the "loopback" interface. For more examples of the syntax see
# the smb.conf man page
;   hosts allow = 192.168.1. 192.168.2. 127.

# if you want to automatically load your printer list rather
# than setting them up individually then you'll need this
printcap name = /etc/printcap
load printers = yes

# It should not be necessary to spell out the print system type unless
# yours is non-standard. Currently supported print systems include:
```

```
# bsd, sysv, plp, lprng, aix, hpux, qnx;
printing = bsd

# Uncomment this if you want a guest account, you must add this to
/etc/passwd
# otherwise the user "nobody" is used
;  guest account = pcguest
# this tells Samba to use a separate log file for each machine
# that connects
log file = /var/log/samba/log.%m
# Put a capping on the size of the log files (in Kb).
max log size = 50

# Security mode. Most people will want user level security. See
# security_level.txt for details.
security = user
# Use password server option only with security = server
;   password server = <NT-Server-Name>

# Password Level allows matching of _n_ characters of the password for
# all combinations of upper and lower case.
;  password level = 8

# You may wish to use password encryption. Please read
# ENCRYPTION.txt, Win95.txt and WinNT.txt in the Samba documentation.
# Do not enable this option unless you have read those documents
;  encrypt passwords = yes
;  smb passwd file = /etc/smbpasswd

# Unix users can map to different SMB User names
;  username map = /etc/smbusers

# Using the following line enables you to customise your configuration
# on a per machine basis. The %m gets replaced with the netbios name
# of the machine that is connecting
;   include = /etc/smb.conf.%m

# Most people will find that this option gives better performance.
# See speed.txt and the manual pages for details
socket options = TCP_NODELAY

# Configure Samba to use multiple interfaces
# If you have multiple network interfaces then you must list them
# here. See the man page for details.
;   interfaces = 192.168.12.2/24 192.168.13.2/24

# Configure remote browse list synchronisation here
# request announcement to, or browse list sync from:
# a specific host or from / to a whole subnet (see below)
;   remote browse sync = 192.168.3.25 192.168.5.255
# Cause this host to announce itself to local subnets here
;   remote announce = 192.168.1.255 192.168.2.44

# Browser Control Options:
# set local master to no if you don't want Samba to become a master
# browser on your network. Otherwise the normal election rules apply
```

```
;   local master = no

# OS Level determines the precedence of this server in master browser
# elections. The default value should be reasonable
;   os level = 33

# Domain Master specifies Samba to be the Domain Master Browser. This
# allows Samba to collate browse lists between subnets. Don't use this
# if you already have a Windows NT domain controller doing this job
;   domain master = yes
# Preferred Master causes Samba to force a local browser election on
startup
# and gives it a slightly higher chance of winning the election
;   preferred master = yes

# Use only if you have an NT server on your network that has been
# configured at install time to be a primary domain controller.
;   domain controller = <NT-Domain-Controller-SMBName>

# Enable this if you want Samba to be a domain logon server for
# Windows95 workstations.
;   domain logons = yes
# if you enable domain logons then you may want a per-machine or
# per user logon script
# run a specific logon batch file per workstation (machine)
;   logon script = %m.bat
# run a specific logon batch file per username
;   logon script = %U.bat

# Where to store roving profiles (only for Win95 and WinNT)
# %L substitutes for this servers netbios name, %U is username
# You must uncomment the [Profiles] share below
;   logon path = \\%L\Profiles\%U

# Windows Internet Name Serving Support Section:
# WINS Support - Tells the NMBD component of Samba to enable it's WINS
Server
;   wins support = yes
# WINS Server - Tells the NMBD components of Samba to be a WINS Client
#     Note: Samba can be either a WINS Server, or a WINS Client, but NOT
both
;   wins server = w.x.y.z

# WINS Proxy - Tells Samba to answer name resolution queries on
# behalf of a non WINS capable client, for this to work there must be
# at least one    WINS Server on the network. The default is NO.
;   wins proxy = yes
# DNS Proxy - tells Samba whether or not to try to resolve NetBIOS names
# via DNS nslookups. The built-in default for versions 1.9.17 is yes,
# this has been changed in version 1.9.18 to no.  dns proxy = no
# Case Preservation can be handy - system default is _no_
# NOTE: These can be set on a per share basis
;  preserve case = no
;  short preserve case = no

# Default case is normally upper case for all DOS files
```

```
;  default case = lower

# Be very careful with case sensitivity - it can break things!
;  case sensitive = no
#============================ Share Definitions ==============================
[homes]
comment = Home Directories
browseable = no
writable = yes

# Un-comment the following and create the netlogon directory for Domain
Logons
; [netlogon]
;  comment = Network Logon Service
;  path = /home/netlogon
;  guest ok = yes
;  writable = no
;   share modes = no

# Un-comment the following to provide a specific roving profile share
# the default is to use the user's home directory;
[Profiles]
;    path = /home/profiles
;    browseable = no
;    guest ok = yes

# NOTE: If you have a BSD-style print system there is no need to
# specifically define each individual printer
[printers]
comment = All Printers
path = /var/spool/samba
browseable = no

# Set public = yes to allow user 'guest account' to print
guest ok = no
writable = no
printable = yes

# This one is useful for people to share files
;[tmp]
;   comment = Temporary file space
;   path = /tmp
;   read only = no
;   public = yes

# A publicly accessible directory, but read only, except for people in
# the "staff" group
;[public]
;   comment = Public Stuff
;   path = /home/samba
;   public = yes
;   writable = yes
;   printable = no
;   write list = @staff

# Other examples.
```

```
## A private printer, usable only by fred. Spool data will be placed in
fred's
# home directory. Note that fred must have write access to the spool
directory,
# wherever it is.
;[fredsprn]
;   comment = Fred's Printer
;   valid users = fred
;   path = /homes/fred
;   printer = freds_printer
;   public = no
;   writable = no
;   printable = yes

# A private directory, usable only by fred. Note that fred requires write
# access to the directory.
;[fredsdir]
;   comment = Fred's Service
;   path = /usr/somewhere/private
;   valid users = fred
;   public = no
;   writable = yes
;   printable = no

# a service which has a different directory for each machine that connects
# this allows you to tailor configurations to incoming machines. You could
# also use the %u option to tailor it by user name.
# The %m gets replaced with the machine name that is connecting.
;[pchome]
;  comment = PC Directories
;  path = /usr/pc/%m
;  public = no
;  writable = yes

# A publicly accessible directory, read/write to all users. Note that all
files
# created in the directory by users will be owned by the default user, so
# any user with access can delete any other user's files. Obviously this
# directory must be writable by the default user. Another user could of
course
# be specified, in which case all files would be owned by that user
instead.
;[public]
;   path = /usr/somewhere/else/public
;   public = yes
;   only guest = yes
;   writable = yes
;   printable = no

# The following two entries demonstrate how to share a directory so that
two
# users can place files there that will be owned by the specific users. In
this
# setup, the directory should be writable by both users and should have the
# sticky bit set on it to prevent abuse. Obviously this could be extended
to
```

```
# as many users as required.
;[myshare]
;   comment = Mary's and Fred's stuff
;   path = /usr/somewhere/shared
;   valid users = mary fred
;   public = no
;   writable = yes
;   printable = no
;   create mask = 0765
```

El archivo smb.conf tiene una estructura simple como se lista:

El archivo está dividido en secciones. Cada sección contiene parámetros, estos definen que comparte SAMBA para exportar.

Una sección global define los parámetros que tienen control general de las características de SAMBA.

Otra sección que comienza con un nombre encerrado entre corchetes (ejemplo [home]) y continua hasta la siguiente sección.

Especifica un parámetro con la siguiente sintaxis:

```
Nombre = valor
```

El nombre puede estar hecho de una o más palabras separadas por espacio. El valor es booleano (falso o verdadero, si o no, 1 o 0); numérico o una cadena de caracteres.

Los comentarios son precedidos de punto y coma (;) y deben ponerse en cada línea.

Las líneas pueden ser continuadas con un slash (\ ) y luego un <enter> y con ello podrá continuar la línea.

Este archivo tiene 3 grandes partes:

Parámetros globales.
La sección del directorio compartido.
La sección de las impresoras compartidas.

Los parámetros globales, hace que pueda compartir, ponen las normas del sistema. La sección [home] y [printer] son instancias especiales de servicio. El término servicios en la terminología de SAMBA de los directorios e impresoras son compartidos o se pueden exportar como clientes de la red. La configuración SAMBA usa un nivel de seguridad de usuario y está especificado en el */etc/smb.conf* por el parámetro `security = user`. Esta seguridad es requerida, ya que el login y contraseña de Windows es muy fácil de suplantar.

3.2.5 Configuración del servidor SAMBA

Se verá en la configuración del servidor SAMBA exportar un directorio público exclusivo SAMBA.

1) Se entra a la computadora y se inicia sesión con Linux.

2) Se hace una copia de respaldo.

```
#cp /etc/smb.conf /etc/smb.comf.orig
```

Se edita el archivo */etc/smb.conf* y se ubica en la sección *[public]*:

```
; [public]
;    comment = Public Stuff
;    path = /home/samba
;    public = yes
;    writable = yes
;    printable = no
```

Se remueve los comentarios, se salva el archivo y debe quedar como sigue:

```
[public]
   comment = Public Stuff
   path = /home/samba
   public = yes
   writable = yes
   printable = no
```

Ahora que se tienen las modificaciones del archivo, pero se necesita que los servicios vean estos cambios ya sea leyendo los anteriores y los tiene cargados usados por el puerto 139. Hay varias formas de que los servicios lean la nueva configuración.

i)Reiniciar la computadora (no es recomendable, pero se usa mucho).

ii)Usar el comando *kill*

iii)Usar el script de los servicios para SAMBA.

3)Ahora se usará el comando `kill`:

```
#kill -HUP `ps ax | grep smbd | awk '{print $1}'`
#kill -HUP `ps ax | grep nmbd | awk '{print $1}'`
```

Con ello se logra que el *daemon* vuelva a leer el archivo */etc/smb.conf* y lea los cambios.

4) Si se usa el script de SAMBA, ubicado en la ruta */etc/rc.d/init.d* para detener, se reinicia los servicios SAMBA y se lean las nuevas modificaciones del archivo */etc/smb.conf*.

```
#/etc/rc.d/init.d/smb stop
#/etc/rc.d/init.d/smb start
```

Si se exportar el directorio de casa del servidor Linux y se puede ver en Windows 95.

5) Editar el archivo */etc/smb.conf* y localizar la sección *[homes]*

```
[homes]
comment = Home Directories
browseable = no
writable = yes
```

Con ello se exportan los homes de los usuarios. Reiniciar los servicios si hubo cambios:

```
#/etc/rc.d/init.d/smb stop
#/etc/rc.d/init.d/smb start
```

Modificar el archivo */etc/smb.conf* para exportar el CDROM y se puedan accesar las computadoras clientes (Windows ó Linux).

6)Editar el archivo */etc/smb.conf* y agregar las nuevas líneas para exportar el CDROM (formato de solo lectura).

```
[cdrom]
comment = Espacio temporal dado al CDROM
path = /mnt/cdrom
read only = yes
public = yes
```

Después de salvar los cambios, reiniciar los servicios y además montar el CDROM en formato de lectura:

```
#mount -r -t iso9660 /dev/cdrom /mnt/cdrom
```

7) Exportando un directorio de dos o mas usuarios. Aquí veremos que hay una forma de agregar dos ó más usuarios con el parámetro "valid users", aquí se verá un ejemplo que compartan el mismo directorio con tres usuarios que estén registrados en el sistema (ejemplo: falcon, kyo y mena).

8)Editemos el archivo */etc/smb.conf*

Se agregan estas líneas:

```
[compartir]
comment = Dir. compartido para el staff
path = /home/staff
valid users = falcon kyo mena
writable = yes
public = no
```

```
printable = no
create mask = 0765
```

Se salvan y se tiene un directorio común para esta tres personas. Reiniciamos servicios de SAMBA.

9)A veces puede equivocarse en escribir una mal sintaxis en el archivo */etc/smb.conf*, una forma de verificarlo es editarlo y buscar error en el archivo (algo tedioso), pero hay un comando que hace esto por el usuario llamado *testparms*. Usa el siguiente comando:

```
#testparms
```

10)También se necesita saber qué usuarios y de dónde se están conectando y se tendrá el siguiente comando:

```
#smbstatus          (Usarlo cuando hay conexión con clientes)
```

El archivo */etc/smb.conf* contiene macros para la versatilidad y flexibilidad del mismo. SAMBA puede sustituir macros de los parámetros de servicios. Para diseñar un macro es necesario crear un símbolo del porcentaje (%) como el primer caracter de alguno de los nombres predefinidos. Como se muestra en el siguiente cuadro:

MACRO	DESCRIPCIÓN
%S	Servicio común o compartido si existe (si existe alguno)
%P	Directorio de root del servicio común o compartido (si existe alguno)
%u	Nombre del usuario del servicio común o compartido (si existe alguno)
%g	Grupo primario del nombre de %u
%U	Sesión del usuario de la respuesta, pero no necesariamente la primera respuesta
%G	Grupo primario del nombre del %U
%H	El directorio de casa del usuario obtenido por %u
%v	Numero de versión de SAMBA
%h	Nombre del host del servidor de SAMBA
%m	Nombre NetBIOS de la computadora cliente
%L	Nombre NetBIOS del servidor SAMBA
%M	Nombre de Internet de la computadora cliente
%d	Proceso ID del servidor común de procesos
%a	Arquitectura del servidor remoto
%I	Dirección IP de la computadora cliente
%T	Dato y tiempo común

11) Ahora si se tiene una computadora con plataforma Windows (3.11, 9x, ó NT) cliente, además del protocolo NetBeui activado, se usa las propiedades de Windows para buscar el dominio correspondiente a la computadora Linux y poder accesar a los recursos compartidos (ya que se verán cuatro servicios activados: home directory, impresoras, cdrom y temporal).

Se hace un resumen del cliente que se utilizó (preferentemente de Windows) y cómo se hace la conexión para iniciar sus recursos en el servidor SAMBA. Se hace una pregunta.

Si se tiene una computadora Linux (cliente) se puede conectar al servidor SAMBA (con Linux) vía el protocolo SAMBA.

3.2.6 Conexión del cliente (Linux) con el servidor vía SAMBA

Si queremos ver la información que se exporta, en vez de un cliente Windows se tiene ahora un cliente Linux, para esto se usa el comando *smbclient* que incluye una breve sintaxis:

```
#smbclient [opciones] servidor

Ejemplo:
#smbclient -L nombre_del_servidor
```

El comando smbclient tiene las siguiente banderas:

-p *port*	Conecta al puerto específico
-d *debuglevel*	El nivel de debuglevel y éste tiene niveles del 0 al 5
-l *logbasename*	*Nombre base* de los archivos logdebug
-n nebios *name*	Usa este nombre como mi nombre de Netbios
-P	Conecta el servicio como una impresora
-M *host*	Manda un mensaje winpopup al host
-L *host*	Obtiene una lista de objetos compartidos disponible en un host
-I dest *IP*	Usa esta IP para conectarse
-U *username*	Pone el nuevo *username* en la computadora destino
-W workgroup	Pone el nuevo *workgroup* en la computadora destino
-t código de terminal	termina la E/S en código
-T <c | x>IXgbnA	Comando en línea de tar
-D *directory*	Inicia desde el directorio

Si se quiere ver el directorio temporal que se está exportando (public) y es algo similar a la conexión tipo Windows que utiliza el protocolo Netbios:

\\servidor\recurso_compartido

13) Esta conexión será:

```
#smbclient \\\\servidor \\recurso_compartido
```

Se pone otro (\) en Linux ya que este símbolo es un caracter especial y se necesita que se lea como un carácter, así que se tiene que proteger (y para esto el carácter especial es él "\").

```
#smbclient \\\\servidor\\public
```

Después preguntará el passwd y dará un prompt de este tipo:

```
smb\>
```

Se pueden usar comandos mixtos tanto los de MS-DOS (dir, copy, md, etc.) como Linux (ls, cp, mkdir, etc.), también se puede hacer transferencia de archivos como el ftp (mencionado en la introducción) y otros comandos mixtos, ya que se puede conectar bajo el protocolo netbios a computadoras Windows y UNIX.

Otra utilería importante es la forma de montaje vía SAMBA con los directorios compartidos y la sintaxis es la siguiente:

```
smbmount //servidor/servicio /punto de montaje [opciones]
```

Si se usa el comando *smbmount -h* se usarán unas opciones como:

-p puerto	Se conecta al puerto
-s nombre_del_servidor	Nombre del servidor NetBios
-c nombre_del_cliente	Nombre del cliente NetBios
-I nombre de la computadora	El hostname de la computadora
-U username	El nombre del usuario
-P password	Se usa este password
-h	Imprime ayuda

```
#smbmount //cerbero/public /mnt -c cliente
```

Si no marca error por el password o porque no está compartido, entonces se verá como resultado la siguiente salida usando el comando *mount*:

```
/dev/hda3 on / type ext2 (rw,usrquota)
none on /proc type proc (rw)
/dev/hdb on /mnt/cdrom type iso9660 (ro)
//cerbero/public on /mnt type smbfs (0)
```

14) También con el comando *smbstatus*:

```
#smbstatus
```

15) Para desmontar directorios vía SAMBA se usará el comando *smbumount*:

```
#smbumount /punto_de_montaje
```

3.3 IP MASQUERADE

3.3.1 Conceptos generales

En el mundo de las comunicaciones y en especial el de la computación tener varias plataformas (ejemplo : Windows, Novell, Unix, Linux, OS/2, VAX, Amiga, MacOs, etc.) en las computadoras implica que dichos sistemas operativos están expuestos a los accesos de intrusos pueden usar recursos y/o destruir la información de las computadoras. Por otro lado, uno de los problemas que enfrenta actualmente Internet es la escasez de direcciones IP ya que día con día se conectan más computadoras a la red mundial para conseguir información, comercio electrónico, etc. por lo que una manera fácil de resolver estos problemas es la implantación de un IP masquerade utilizando una computadora con sistema operativo Linux, ya que con esto podemos conectar redes LAN usando dirección IP reservadas o sin clase y las computadoras protegidas tengan un acceso invisible al Internet y no estén expuestas a intrusos, además se ahorra un número considerable de direcciones IP, ya que solamente se usará la IP oficial en la computadora que se funcionarán como gateway(20).

La computadora con Linux hará su función de gateway y su respectiva labor de filtrado de paquetes provenientes de Internet. Para entender un poco de donde se puede usar direcciones IP reservadas ó sin clase, veremos como están distribuidas estas mismas.

Cada host y enrutador de Internet tiene una dirección IP, que codifica sin número de red y su número de host. La combinación es única ; no hay dos computadoras que tengan la misma dirección IP. Todas las direcciones de IP son de 32 bits de longitud y se usan en los campos de dirección origen y de dirección destino. Donde hay cuatro clases principales de dirección : A, B, C y D.

En una dirección de clase A, el primer byte representa la porción de red y los otros tres identifican al host. Esto significa que la red puede tener millones de hosts, debido a que se dispone de 24 bits para identificar la dirección del host. De hecho, la dirección 0.0.0.0 (conocida como el ruteo por default) y 127.0.0.0 (la red de retorno de lazo) tiene un especial significado y no está disponibles para su identificación de la red. Así éstas son únicamente de la dirección 126 que esta disponible a la clase A.

La dirección de clase B tiene16 bits para la red y otros 16 para el host. El rango de la clase B va de la 128 a la 191, cada red contiene arriba de 32766 posibles interfases.

La clase C tiene 24bits para la red y 8 para el host, la clase B tiene un rango de 192 a 254. Estas así son 4,194,303 disponibles números de red de la clase C, cada red contiene 254 interfases.

También existen unas direcciones especiales que están reservadas para la "no conexión" en la Internet, estas direcciones son:

1 red Clase A	10.0.0.0
16 redes de Clase B	172.16.0.0-172.31.0.0
256 redes Clase C	192.168.0.0-192.168.255.0
x.x.x.0	Identifica a toda la red.
x.x.x.255	A todos los hosts de la red especifica

Todas las direcciones de difusión de host para todas las redes actuales.

También existe la máscara de red, con ella se puede hacer saber si es una red de tipo A, B, C o si se quiere que una clase tipo C se divida en subredes, con el movimiento de bits en la mascara de red podemos dividirlo.

La máscara de red estándar es todos los bits de red en una dirección puesta a 1 y todos los bits del host puesta a 0. Estos son los estándares de las máscara de red de las tres clases :

Clase A máscara de red : 255.0.0.0

Clase B máscara de red : 255.255.0.0

Clase C máscara de red : 255.255.255.0

Clase D máscara de red : 255.255.255.255

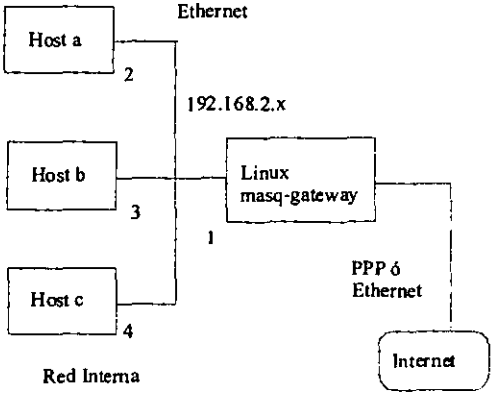
Ahora se verá la aplicación de las subredes con movimiento de los bits en la máscara de red :

No. de Subred	No. de host en la red	Mascara de red
2	126	255.255.255.128
4	62	255.255.255.192
8	30	255.255.255.224
16	14	255.255.255.240
32	6	255.255.255.248

Otra manera de ver la división de la subred es las siguiente :

Mascara de Red	subred	Red	Broadcast	min. IP	Máx. IP	Host	Total Host
128	2	0	127	1	126	126	
		128	255	129	254	126	252
192	4	0	63	1	62	62	
		64	127	65	126	62	
		128	191	129	190	62	
		192	255	193	254	62	248
224	8	0	31	1	30	30	
		32	63	33	62	30	
		64	95	65	94	30	
		96	127	97	126	30	
		128	159	129	158	30	
		160	191	161	190	30	
		192	223	193	222	30	
		224	255	225	254	30	240

Ahora se dará un ejemplo de la forma de conexión que se pretende hacer. Aquí se sabe que se tiene una red interna con una variedad de sistemas operativos y que todas esas computadoras quieren usar Navegadores WEB, ftp, telnet, Real Audio, IRC, etc.



Como se puede observar se tiene una red interna, que puede tener cualquier sistema operativo (Windows9x/NT, Windows 3.11, Novell, Unix, Macintosh MacTCP, Linux, etc.) donde se ve que se usa una red clase tipo C (192.168.2.0) y el host con Linux tiene una dirección 192.168.2.1 y otra dirección que sí está reconocida por la red mundial, además esta computadora tiene un sistema masq-gateway que convierte a todas estas conexiones y tiene la capacidad de pasar la información a las computadoras de la Internet con la red interna y la distribución de la información. Como se puede observar el kernel de Linux tiene un mecanismo para el uso de un firewall : masquerade de paquetes IP. Así, la dirección IP fuente es reemplazada por la dirección IP local y el puerto fuente es reemplazado por un puerto local. Al iniciar una administración es guardada de sesiones enmascaradas (masqueraded), después los paquetes que regresan de los puertos serán automáticamente "desenmascarados" y mandados al sistema que originalmente hizo la petición :

Si se quiere hacer una conexión telnet (puerto 23), de la computadora 192.168.2.2 (computadora interna) a la computadora 132.248.52.2 (computadora Externa), pasando por un sistema Linux enmascarado (masquerading) con IP 192.168.2.1-132.248.59.98

	Fuente		Destino	
	Dirección IP	Puerto	Dirección IP	Puerto
paquete original	192.168.2.2	1027	132.248.59.2	23
Enmascarar	192.168.2.1/132.248.59.98	32595	132.248.59.2	23
paquete retornado	132.248.59.2	23	192.168.2.1/132.248.59.98	23595
Desenmascarar	132.248.59.2	23	192.168.2.2	1027

Aquí hay una cuestión interesante, cómo está haciendo el intercambio de paquetes una computadora que tiene como dirección IP una de las reservadas para no tener conexión, y se pueda mandar a otra que sí tiene conexión al exterior. Esto es muy simple si recordamos que en una computadora con Linux se tienen que poner dos tarjetas de red ó dos conexiones con modem o la combinación de ambos, una de estas conexión es estará con la red interna y la otra con la externa, mientras que el kernel de Linux tiene programas que hace el intercambio de paquetes entre las dos tarjetas y todo es transparente, claro que se tiene que hacer una serie de pasos para obtener la configuración deseada, pero con ello se logran tres cosas muy importantes :

i) Proteger de intrusos a la red interna.

ii) Ahorrar direcciones IP y distribuirlas a otras redes de mayor importancia o utilizar otros masquerade.

iii) Administrar y monitorear las conexiones de los usuarios, ya que con esto se sabe que computadora se conecta, qué protocolo y cuánto dura su conexión.

3.3.2 Instalación y configuración del hardware

1) Configurar las dos tarjetas de red, o conexión por módem (en este caso manejaremos dos tarjetas, para la congruencia del texto).

2) Se Recompila el kernel con las opciones de soporte para:

```
#make menuconfig

*Buscar las 2 tarjetas correspondientes para tener sus
drivers.
```

*Prompt for development and/or incomplete code/drivers
CONFIG_EXPERIMENTAL
-this will allow you select experimental ip_masq code compiled into the kernel.

*Enable loadable module support
CONFIG_MODULES
-allows you to load modules

*Networking support
CONFIG_NET

*Network firewalls
CONFIG_NET

*TCP/IP networking
CONFIG_INET

*IP: forwarding/gatewaying
CONFIG_IP_FORWARD
*IP: firewalling
CONFIG_IP_FIREWALL

*IP: masqueradinf (EXPERIMENTAL)
CONFIG_IP_MASQUERADE
-although it is experimental, it is a "MUST"

*IP: always defragment
CONFIG_IP_ALWAYS_DEFRAG
-highly recommended

*Dummy net driver support
CONFIG_DUMMY
-recommended

Después de compilar el kernel, también compilaremos e instalaremos los módulos:

```
#make modules ; make modules_install
```

Mientras se compila el kernel y módulos se puede configurar la nueva interfaz copiando el archivo /etc/sysconfig/network-scripts/ifcfg-eth0:

```
#cp /etc/sysconfig/network-scripts/ifcfg-eth0
/etc/sysconfig/network-scripts/ifcfg-eth1
```

Ahora se modifica para soportar la red interna se queda de esta manera:

```
#vi /etc/sysconfig/network-scripts/ifcfg-eth1

DEVICE=eth1
IPADDR=192.168.2.1
NETWORK=192.168.2.0
NETMASK=255.255.255.0
BROADCAST=192.168.2.255
ONBOOT=yes
BOOTPROTO=none
```

También se cargarán los módulos cada vez que inicie la computadora los paquetes del conocido ipv4 (ejemplos : ip_masq_ftp, ip_masq_raudio, ip_masq_irc, etc.). Para que se puedan usar esos servicios en forma transparente, se ponen estas líneas en el archivo */etc/rc.d/rc.local* y se ejecutará cada vez que se inicie la computadora:

```
#vi /etc/rc.d/rc.local
      .
      .
      .
/sbin/depmod -a
/sbin/modprobe ip_masq_ftp
/sbin/modprobe ip_masq_raudio
/sbin/modprobe ip_masq_irc
```

Se salva el archivo.

Al terminar de recompilar el kernel, reiniciar la computadora, se tendrán que reconocer las dos tarjetas, los módulos y posteriormente se ordenará a una de ellas que va estar en una red interna y la otra a la red mundial.

Nota: Para ver si reconocen las tarjetas se usa el comando *dmesg*, para la configuración de las tarjetas de red se usa *ifconfig*, para los módulos *lsmod* o ver el archivo */var/log/messages*

```
#dmesg
        .
        .
        .
        .
PPP Dynamic channel allocation code copyright 1995 Caldera, Inc.PPP line discipline registered.
tulip.c:v0.88 4/7/98 becker@cesdis.gsfc.nasa.gov
eth0: Digital DS21142/3 Tulip at 0x6100, 00 48 54 00 24 61, IRQ 11.
eth0:  EEPROM default media type Autosense.
eth0:  Index #0 - Media 10baseT (#0) described by a 21142 Serial PHY (2) block.
eth0:  Index #1 - Media 10baseT-FD (#4) described by a 21142 Serial PHY (2) block.
eth0:  Index #2 - Media 100baseTx (#3) described by a 21143 SYM PHY (4) block.
eth0:  Index #3 - Media 100baseTx-FD (#5) described by a 21143 SYM PHY (4) block.

eth1: 3c509 at 0x300 tag 1, BNC port, address 00 a0 24 2f 30 69, IRQ 10
3c509.c:1.07  6/15/95 becker@cesdis.gsfc.nasa.gov
Partition check:
        .
        .
```

#ifconfig

```
lo   Link encap:Local Loopback
        inet addr:127.0.0.1 Bcast:127.255.255.255 Mask:255.0.0.0
        UP BROADCAST LOOPBACK RUNNING MTU:3584  Metric: 1
        RX packets:307 errors:0 dropped:0 overruns:0
        TX packets:7  errors:0 dropped:0 overruns:0

eth0  Link encap:Ethernet Hwaddr 00:48:54:00:24:61
        inet addr:132.248.59.66 Bcast:132.248.59.255 Mask:255.255.255.0
        UP BROADCAST RUNNING MULTICAST MTU:1500 Metric: 1
        RX packets:1546 errors:2 dropped:0 overruns:0
        TX packets:2347 errors:0 dropped:0 overruns:0
        Interrupt:11  Base Address: 0x6100

eth1  Link encap:Ethernet Hwaddr 00:A0:24:2F:30:69
        inet addr:192.168.2.1 Bcast:192.168.2.255 Mask:255.255.255.0
        UP BROADCAST RUNNING MULTICAST MTU:1500 Metric: 1
        RX packets:2034 errors:2 dropped:0 overruns:0
        TX packets:12247 errors:0 dropped:0 overruns:0
        Interrupt:10  Base Address:0x300
```

#lsmod

```
Module          Pages    Used by
ip_masq_raudio      1         0
ip_masq_ftp         1         0
ip_masq_irc         1         0
sb                  6         0
uart401             2    [sb]  0
sound              16    [sb uart401]      0
```

3.3.3 Configuración de varios sistemas operativos

Ahora se hará la configuración de las computadoras dentro de la red virtual de los siguientes sistemas operativos :

Windows9x. Después de instalar la tarjeta y de que la reconozca, ir al "*panel de control/red*" y agregar el "*protocolo de TCP/IP*".

En "*TCP/IP propiedades*", colocar en "*IP Address*" y poner la dirección IP del rango 192.168.2.x (1<x<255) y la máscara de subred de 255.255.255.0. Se agrega 192.168.2.1 como gateway en la opción de "gateway".

Después se agregará en la "*Configuración del DNS/DNS Server search order*" poner un DNS conocido.

Al terminar de configurar estos cambios, reiniciar la computadora correspondiente y se puede verificar si la computadora ve al servidor Linux, usando el comando ping en la parte de "Ejecutar" se pondrá : *ping 192.168.2.1.* Si responde la computadora es que se está realizando gran parte de la configuración.

Windows for Workgroup 3.11. Aquí hay dos formas de configurar:

Instalar el paquete de TCP/IP 32b, posteriormente en "Main/Windows setup/Network setup", dar un click en "Drivers". Poner la dirección IP 192.168.2.x (1<x<255), también la subred de 255.255.255.0 y default gateway 192.168.2.1.

Los drivers de la tarjetas y emuladores deben estar configurados (winpkt), posteriormente en el autoexec.bat se pondrá la sintaxis **set ip= 192.168.2.x. (1<x<255)** Ahora en el programa Trumpet Winsock en el setup se pondrán los siguientes valores :

IP Address 192.168.2.x (1<x<255)
Netmask 255.255.255.0
Domain Name Server
Default Gateway 192.168.2.1

Se reinicia el Trumpet y estará los nuevos valores de configuración.

Windows NT 4.0. Se ubica en el panel de control, se selecciona red, posteriormente se observa el módulo de protocolos, se selecciona la configuración de TCP/IP y se introducen los valores :

dirección IP 192.168.2.x (1<x<255)
netmask 255.255.255.0.
Gateway 192.168.2.1

Configuración de Sistemas UNIX. Si no se tiene una tarjeta de red, se tiene que configurar recompilando el kernel, posteriormente se instala las utilerías de TCP/IP. Dirección IP 192.168.2.x (1<x<255), Gateway 192.168.2.1 y netmask 255.255.255.0 como también el BRADCAST 192.168.2.255. Un ejemplo de Linux redhat está en el archivo */etc/sysconfig/network-scripts/ifcfg-eth0. Se* agrega el DNS en el archivo */etc/resolv.conf*, se actualiza el archivo */etc/networks* y se restaura los servicios o simplemente restaura la computadora.

3.3.4 Configuración del gateway-masquerade

Se usa el comando ping : **ping 192.168.2.1** y se observa si el gateway está funcionando.

La computadora Linux usará el programa de ruteo entre las dos tarjetas para que haga una función de gateway, llamado este programa en Linux **ipfwadm** (ip-firewall-administrator), aunque tiene la segunda opción cuando se usa el comando **ipchains.**

Las políticas de IP Forwarding de la opción **ipfwadm** y teniendo en orden tanto el kernel, las tarjetas y módulos. Ahora necesitamos el gateway-real, DNS, para la computadora Linux.

Ahora, las opciones de **ipfwadm** para mandar los paquetes apropiados en la computadora gateway :

```
#ipfwadm -F -p deny
#ipfwadm -F -a m -S yyy.yyy.yyy.yyy/x -D 0.0.0.0/0
```

Con **ipchains** :

```
#ipchains -P forward DENY
#ipchains -A forward -j MASQ -s yyy.yyy.yyy.yyy/x -d 0.0.0.0/0
```

Donde *x* es uno de los siguientes números de acuerdo con la clase de subred y la dirección de la red es *yyy.yyy.yyy.yyy* :

Mascara de red	x	Subred
255.0.0.0	8	Clase Tipo A
255.255.0.0	16	Clase Tipo B
255.255.255.0	24	Clase Tipo C
255.255.255.255	32	Punto a Punto

```
#ipfwadm -F -p deny
#ipfwadm -F -a m -S 192.168.2.0/24  -D  0.0.0.0/0
```

Con **ipchains** :

```
#ipchains -P forward DENY
#ipchains -A forward -j MASQ -s 192.168.2.0/24 -d 0.0.0.0/0
```

En el ejemplo se observa que rutea la red tipo C (192.168.2.0) y no importa a qué puerto se conecte a la salida de la tarjeta de red (0.0.0.0), para no utilizar estos comandos cada vez que se inicie la computadora, se puede mandar a un archivo (ejemplo: /etc/rc.d/rc.local) para que se inicien.

Una de las virtudes de este comando **ipfwadm** es la posibilidad de rutear computadoras punto a punto (de manera individual). En el siguiente ejemplo solamente tendrán acceso a Internet las computadoras con IP 192.168.2.4 y 192.168.2.5 :

```
#ipfwadm -F  -p deny
#ipfwadm -F  -a m -S 192.168.2.4/32  -D  0.0.0.0/0
#ipfwadm -F  -a m -S 192.168.2.5/32  -D  0.0.0.0/0
```

Con **ipchains** :

```
#ipchains -P forward DENY
#ipcahins -A forward -j MASQ -s 192.168.2.4/32  -d 0.0.0.0/0
#ipcahins -A forward -j MASQ -s 192.168.2.5/32  -d 0.0.0.0/0
```

Para observar las diferentes conexiones de las computadoras clientes se puede utilizar la opción de :

```
#ipfwadm -M -l
```

Con **ipchains** :

```
#ipchains -M -L
```

Mostrando algo semejante :

IP masquerading entries

prot	expire	source	destination	ports
tcp	12 :25.1	192.168.2.6	www.unam.mx	1017 (80) ->http
tcp	14 :52.12	rush.comp.fi	cronos.fi-b.unam.mx	1345 (6023) ->telnet

Esta información se encuentra en un ***pseudo-archivo*** llamado /proc/net/ip_masquerade, que se puede leer con la conversión del **ipfwadm** y para **ipchains** es el mismo. Con esto podemos conectar una red y utilizar Internet con una dirección IP, funcionando como gateway el host Linux.

3.4 Firewalls

Un firewall es un dispositivo que previene el acceso a la red local. Este dispositivo es comúnmente un ruteador o una computadora standalone corriendo filtrado de paquetes(2).

Un firewall es básicamente un dispositivo de protección. Lo primero que hay que conocer, es lo que se quiere proteger. Al conectarse a Internet, pone en riesgo tres cosas :

- datos,
- recursos y
- reputación.

3.4.1 Tipo de ataques

Hay muchos tipos de ataques contra los sistemas y muchas formas de clasificarlos. Se dividen generalmente los ataques en tres categorías básicas :

- Intrusión,
- negación de servicio y
- robo de la información.

3.4.1.1 Intrusión

Los ataques mas comunes a los sistemas son las intrusiones; con ellas las personas pueden utilizar sus computadoras. La mayoría de los atacantes quieren utilizar sus computadoras como si fueran usuarios legítimos. Los intrusos cuentan con decenas de formas para tener acceso. Van desde ataques de manipulación social, como por ejemplo, encuentra la forma de saber un puesto importante para la compañía; llaman a un administrador del sistema, diciendo que son esa persona y que necesitan cambiar su contraseña, en ese preciso instante, para hacer un trabajo urgente. Adivinación, prueban combinaciones con el nombre y contraseña de la cuenta hasta que funciona, pasando por formas compiladas de entrar, sin necesidad de saber el nombre y la contraseña de la cuenta.

Los firewalls ayudan a evitar intromisiones de varias maneras. Lo ideal es obstaculizar todas las formas de entrar al sistema sin saber el nombre y la contraseña de la cuenta. Si están configurados de manera correcta, reducen el número de cuentas a que tienen acceso desde el exterior y que, por lo tanto, son vulnerables a que las adivinen o a la manipulación social.

3.4.1.2 Negación del servicio

Un ataque de Negación de Servicios (DoS "Denial of Services") es aquel que esta dirigido en su totalidad a evitar que usted utilicen sus propias computadoras. Un ataque común de

Negación es inundar la red, un intruso es más hábil, también puede inhabilitar los servicios, volverlos a enrutar o reemplazarlos.

Es casi imposible evitar todos los ataques de negación de servicio.

3.4.1.3 Robo de información

Algunos tipos de intromisiones permiten que el atacante obtenga información sin tener que utilizar directamente sus computadoras. Por lo general, estos ataques se aprovechan de los servicios de Internet, que tienen como fin proporcionar información, haciendo que den más información de lo que era su intención, o dándosela a personas equivocada. Muchos servicios de Internet están diseñados para usarlos en Redes de Área Local y no tienen el tipo o grado de seguridad que permitiría emplearlos de manera segura a través de Internet.

La mayoría de las personas que roban información, intentan tener acceso a sus computadoras ; buscan nombres de usuarios y contraseñas. Para fortuna de ellos y desgracia de los demás, es el tipo de información más fácil de obtener al intervenir una red. La información de nombre de usuario y contraseña pasa de manera predecible al principio de muchas interacciones de la red y puede volver a utilizarse de la misma forma.

Las intervenciones de la red usualmente llamadas **sniffers** (analizadores), son muy efectivas para encontrar las contraseñas, pero rara vez se utilizan para recabar otros datos. Obtener la información mas especifica, requiere de una dedicación y paciencia extremas, o el conocimiento de datos deseados pasaran de manera fidedigna por un lugar especifico en un momento dado.

Intervenir redes es mucho más fácil que una línea de teléfono. En las tecnologías más comunes de redes, como Ethernet y Token Ring, cualquier computadora que esta en una Red de área local es capaz de ver todo el tráfico que pasa por la red. En el tráfico que cruza Internet puede pasar un sin número de Redes de Arrea Local, cualquiera de las cuales puede ser un punto de riesgo. Los proveedores de servicios de red y los sistemas de acceso público son blancos más comunes para intrusiones; analizadores colocados ahí pueden tener mucho éxito porque hay mucho tráfico que pasa a través de estas redes.

Existen varios tipos de protección contra el robo de información. Un firewall configurado correctamente lo protegerá contra personas que intenten obtener más información de la que usted pueda dar.

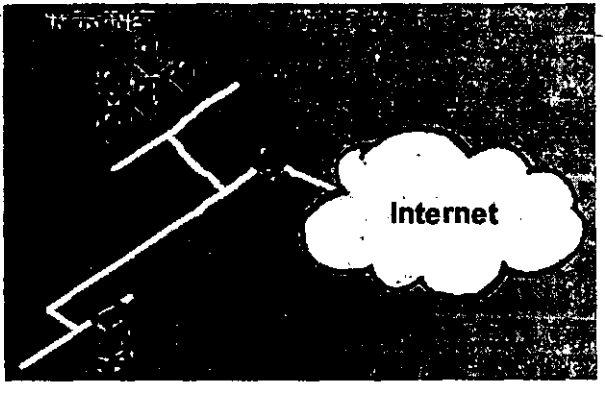
Sin embargo, una vez que se decide dar información a través de Internet, es muy difícil protegerla para que no llegue a un publico no deseado, ya sea a través de una falsa autenticidad o a través de los analizadores. Aunque estos riesgos quedan fuera de la protección que pueda dar un firewall (porque ocurren una vez que intencionalmente se ha permitido que la información fluya fuera de su red).

3.4.2 Arquitectura de firewalls

3.4.2.1 Arquitectura de anfitrión con doble acceso

Una arquitectura de anfitrión con doble acceso se construye alrededor de una computadora anfitrión que tiene por los menos dos interfaces de acceso. Tal anfitrión podría actuar como un enrutador entre las redes que están conectadas sus interfaces; es capaz de enrutar paquetes de IP de una red a otra. Sin embargo, al implementar una arquitectura de firewalls del tipo anfitrión con doble acceso, desactiva esta función de enrutamiento. Así, los paquetes IP de una red no se enrutan en forma directa a la otra red(3). Los sistemas dentro del firewall pueden comunicarse con el anfitrión con doble acceso, pero estos no pueden comunicarse entre sí de manera directa. El tráfico IP entre ellos está bloqueado en su totalidad. La arquitectura de red para un firewall de anfitrión con doble acceso es bastante sencilla:

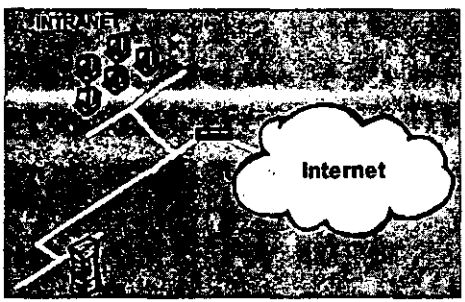
El anfitrión se conecta a Internet y se coloca en una red Interna:



Los anfitriones con doble acceso pueden proporcionar un alto nivel de control. Si no permiten que los paquetes pasen entre las redes internas y externas, pueden tener la seguridad de que con cualquier paquete de la red interna que tenga una fuente externa es evidencia de algún tipo de sistema de seguridad. En algunos casos, un anfitrión con doble acceso permitirá que rechace conexiones que se pretenden para ser un servicio especifico, pero que, en realidad, no contienen el tipo de datos correctos (un sistema de filtrado de paquetes, por otro lado, tiene dificultades con este nivel de control.). Además, la mayoría de los usuarios encuentran que no es conveniente emplear un anfitrión con doble acceso e iniciar una sesión con el antes de salir a Internet.

3.4.2.2 Arquitectura de anfitrión de protección

Mientras una arquitectura de anfitrión con doble acceso proporciona servicios desde un anfitrión conectado a varias redes (pero con el enrutamiento desactivado), una arquitectura de anfitrión de protección proporciona servicios de un anfitrión conectado solo a la red interna, utilizando un enrutador independiente. En esta arquitectura, la seguridad principal se proporciona al filtrado de paquetes (El filtrado evita que las persona evadan los servidores proxy para ser conexiones directas).



El anfitrión bastión esta colocado en la red interna. El filtrado de paquetes en el ruteador de protección esta configurado de tal manera que el anfitrión bastión es el único sistema en la red interna con el que los anfitriones en Internet pueden abrir conexiones (por ejemplo, para entrega de correo electrónico). Aún así solo están permitidos ciertos tipos de conexiones. Cualquier sistema externo que intente tener acceso a los sistema o a los servicios internos, tendrá que conectarse con este anfitrión. Por lo tanto, el anfitrión bastión debe mantener un alto nivel de seguridad.

El filtrado de paquetes también permite que el anfitrión bastión abra conexiones permitidas al mundo exterior (lo permitido lo determina la política de seguridad especifica de su sitio).

La configuración para el filtrado de paquetes en el enrutador de protección se puede hacer mediante una de las siguientes tareas:

Aceptar que otros anfitriones internos abran conexiones con otros en Internet para ciertos servicios.

No se deben permitir todas las conexiones de anfitriones internos.

Poder mezclar e igualar estos enfoques para diferentes servicios; algunos se permiten directamente por medio del filtrado de paquetes; otros se permiten solo de manera indirecta por medio de proxy's; todo depende de la política especifica que el sitio intente cumplir.

Debido a que esta arquitectura permite que los paquetes se muevan de Internet a las redes internas, puede parecer mas riesgoso que la arquitectura de anfitrión con doble acceso, diseñada para que ningún paquete externo alcance la red interna. Sin embargo, en la práctica, la arquitectura de anfitrión con doble acceso también es propensa a fallas que permite que, en realidad, pasen los paquetes de la red externa a la Interna (debido a que esta clase de falla es inesperada es poco probable que existan protecciones contra este tipo de ataques). Además, es mas fácil defender un enrutador que proporcione un conjunto limitado de servicios, que defender un anfitrión. Para casi todos los propósitos, la arquitectura de anfitrión de protección proporciona mayor seguridad y mejor uso que la arquitectura de anfitrión con doble acceso.

Sin embargo en comparación con otras arquitecturas, como la de subred de protección, tiene algunas desventajas. La principal es que si un atacante logra penetrar al anfitrión bastión, no queda nada en la ruta de seguridad de la red entre ese anfitrión y el resto de los anfitriones internos. El enrutador también presenta un solo punto de falla; si este se haya en peligro, toda la red estará a merced de un atacante. Por esta razón, la arquitectura de subred de protección se ha vuelto más popular.

3.4.2.3 Arquitectura de subred de protección

La arquitectura de subred de protección agrega una capa adicional de seguridad a la arquitectura de anfitrión de protección al añadir una red de perímetro que aísla aún mas la red interna de Internet.

Por su naturaleza, los anfitriones bastión son las computadoras mas vulnerables de su red. A pesar de sus mejores esfuerzos por protegerlas, son las computadoras que con mayor probabilidad sean atacadas, pues son las que pueden ser atacadas. Si, al igual que una arquitectura de anfitrión de protección, su red interna está totalmente abierta para ser atacada desde su anfitrión bastión, entonces este es un blanco muy tentador. No hay alguna otra defensa entre el y sus computadoras internas (a parte de cualquier seguridad de anfitrión que pueda tener, por lo general es muy poca.).

Al aislar al anfitrión bastión en una red de perímetro, puede reducir el impacto de una entrada forzada a él. Le da al intruso un poco de acceso, pero no del todo.

Con la forma mas sencilla de arquitectura de subred de protección, hay dos enrutadores de protección, cada uno conectado a la red de perímetro. Uno colocado entre la red de perímetro y la red interna, y el otro entre la red perímetro y la red externa (por lo general, Internet). Para entrar a la red interna con este tipo de arquitectura, un atacante tendría que penetrar ambos enrutadores. Aunque el atacante lograra de alguna forma penetrar al anfitrión bastión, aun tendría que pasar al enrutador interior. No hay un punto vulnerable único que ponga en riesgo la red interna.

Algunos sitios van tan lejos como para crear una serie de capas de redes perímetro entre el mundo exterior y su red interna. Los servicios menos confiables y más vulnerables se colocan en las redes de perímetro exteriores, más lejos de la red interior. La idea es que el atacante que penetre la computadora de la red de perímetro exterior, le costará más trabajo atacar con éxito la computadoras internas debido a las capas adicionales de seguridad entre el perímetro exterior y la red interna. Sin embargo, esto solo es cierto si las distintas capas tienen sentido; si los sistemas de filtrado entre cada capa permiten lo mismo entre todas las capas, las capas adicionales no proporcionan mayor seguridad.

3.4.3 Ventajas y desventajas de un firewall

3.4.3.1 Ventajas

Los firewalls pueden hacer mucho por la seguridad de su sitio. De hecho, algunas de sus ventajas se extienden más a la seguridad, como lo describimos a continuación.

El firewall es un cuello de botella, todo el tráfico que entra y sale debe pasar por este estrecho punto de inspección. Un firewall da un grado de eficiencia enorme a la seguridad de redes porque le permite concentrar sus medidas de seguridad en este punto de inspección: el punto donde su red se conecta con Internet.

Muchos de los servicios que las personas demandan de Internet son, por su propia naturaleza, inseguros. Un firewall es como un agente de tránsito para estos servicios. Refuerza las políticas de seguridad del sitio, permitiendo que pasen sólo los servicios "aprobado" y aquellos que cumplen con las reglas establecidas para ello.

Por ejemplo, la administración de un sitio puede decidir que ciertos servicios, como el Sistema de Archivos de Red de SUN (NFS o Network File System) y los servicios de Información para Redes (conocido con anterioridad como Yellow Pages, NIS/YP), son simplemente muy riesgosos para utilizarlos a tráves de un firewall. No importa que sistema intente ejecutarlos o que usuario los quiera: el firewall los mantendrá potencialmente peligrosos dentro de él.

Debido a que todo transita por el Firewall, éste proporciona un buen lugar para reunir información sobre el uso de sistemas y redes (o el mal uso). Como punto único de acceso, un firewall puede registrar lo que ocurre entre la red protegida y la red externa.

3.4.3.2 Desventajas

Los firewalls ofrecen excelente protección contra las amenazas a la red, pero no son una solución de seguridad total. Ciertas amenazas están fuera del control del firewall. Se debe encontrar otras formas de protegerse contra ellas incorporando seguridad física, seguridad para anfitrión y educación para el usuario en su plan general de seguridad. A continuación se analizan algunas de las desventajas de los firewalls.

Un firewall puede evitar que un usuario del sistema envíe información del propietario fuera de la organización a través de su conexión de red; también lo evitaría no teniendo una conexión de red. Pero ese mismo usuario podría copiar los datos en disco, cinta o papel y sacarlos del edificio en su portafolio.

Un firewall puede controlar el tránsito que pasa por él de manera eficaz; sin embargo, no hay nada que pueda hacer con el tránsito que no pasa por él. Como una línea conmutada de teléfono por atrás del firewall.

Un firewall esta diseñado para proteger de amenazas conocidas y novedosas. Sin embargo ningún firewall puede defenderse de manera automática contra cada amenaza nueva que surga. Periódicamente, las personas descubren nuevas formas de atacar, utilizando servicios que antes eran confiables o usando ataques que simplemente no se le habían ocurrido a nadie. No se puede instalar un firewall una vez y esperar que lo proteja para siempre.

Los firewalls no pueden mantener los virus de PC y Macintosh fuera de la red. Aunque muchos firewall revisan el tránsito que entra para determinar si pueden pasar a la red interna, la exploración ocurre en su mayoría en el nivel direcciones fuente y destino y números de puertos, no en los detalles de los datos. Aún cuando se tenga filtrado de paquetes o software proxy complejo, la protección contra el virus en un firewall no es muy práctico. Hay demasiado tipo de virus e infinidad de formas en que uno de ellos puede ocultarse dentro de los datos.

Detectar un virus al azar en un paquete de datos que pasa a través de un firewall es muy difícil porque requiere de:

- Reconocer que el paquete es parte de un programa.
- Determinar cómo debe verse el programa.
- Determinar que el cambio se deba a un virus.

Incluso el primer punto es un reto. La mayor parte de los firewalls son computadoras protectoras de varios tipos con diferentes formatos de archivos ejecutables. Un programa puede ser un archivo ejecutable compilado un script y muchas computadoras que soportan múltiples archivos ejecutables compilados. Además, la mayoría de los programas están

empaquetados para su transporte, y con frecuencia también están comprimidos. Los paquetes transferidos por medio de correo electrónico o noticias de Usenet también están codificados a ASCII de diferentes maneras.

3.4.4 Postura de falla segura

Otro principio fundamental es que, en la medida de lo posible, los sistemas deben tener una falla segura, es decir, si van a fallar deben hacerlo de tal forma que nieguen el acceso a un atacante en lugar de dejarlo entrar. La falla también puede causar la negación de acceso a usuarios legítimos hasta que se hagan reparaciones, pero por lo general es algo aceptable.

Las fallas seguras son otro principio de amplia aplicación en lugares familiares en nuestra vida diaria. Los dispositivos eléctricos están diseñados para apagarse - detenerse - cuando fallan de alguna forma.

La aplicación más importante de este principio de seguridad para redes reside en seleccionar una postura de su sitio respecto a la seguridad. Su postura, es en esencia, la actitud general de su sitio en relación con su seguridad.

Hay dos posturas fundamentales que puede adoptar con respecto a decisiones y políticas de seguridad:

- Postura de negación prestablecida: especifique lo que permite y prohíba todo lo demás.
- Postura de permiso prestablecida: especifique solo lo que prohíbe y permita todo lo demás.

3.4.4.1 Postura de negación prestablecida

Lo que no está permitido expresamente está prohibido. La postura de negación prestablecida tiene sentido desde el punto de vista de la seguridad porque es una postura de falla segura, acepta lo que no puede dañar. Es la opción obvia más segura para la mayoría de las personas pero en lo general no lo es para los usuarios.

Con la postura de negación establecida, se prohíbe todo por omisión; después, para determinar lo que se va a permitir debe:

- Examinar los servicios que necesitan los usuarios.
- Considerar como afectaría tales servicios la seguridad y como proporcionarlos de manera segura.
- Permitir solo los servicios que comprende, que puede proporcionar con seguridad y para los cuales ve una necesidad legitima.

Los servicios se activan basándose en cada caso. Se empieza por analizar la seguridad de un servicio especifico y se hace un balance comparando los efectos que tendría en la seguridad contra las necesidades de los usuarios basándose en este análisis y en la disponibilidad de varias soluciones para mejorar la seguridad del servicio.

Un servicio se puede proporcionar a todos los usuarios con toda seguridad mediante el filtrado de paquetes o con los sistemas Proxy, disponibles fácilmente en el mercado.

3.4.4.2 Postura de permiso prestablecido.

Lo que no esta prohibido expresamente esta permitido. La mayoría de los usuarios y administradores prefieren la postura de permiso prestablecido. Tienden a suponer que todo estará, por omisión, permitido y se irán prohibiendo ciertas acciones y servicios problemáticos específicos conforme sean necesarios.

Primero, se conoce de antemano y de manera precisa cuales son los peligros específicos, como explicarlos para que los usuarios los comprendan y como protegerse contra ellos. Adivinar que peligros podrían estar en el sistema o en Internet es, en esencia una tarea imposible.

Segundo, la postura de permiso prestablecido, tiende a degenerar en una carrera armamentista que continua creciendo entre quienes dan mantenimiento al firewall y a los usuarios. Quien da mantenimiento prepara las defensas contra la acción o la inacción del usuario; los usuarios inventan nuevas formas fascinantes e inseguras de hacer las cosas; y este proceso se repite una y otra vez. Quien da mantenimiento siempre intenta mantenerse al día es inevitable que existan periodos de vulnerabilidad entre el momento que se instala un sistema, el momento que se descubre un problema de seguridad y el momento en que la persona que le da mantenimiento pueda responder al problema.

Casi las únicas personas que se benefician de la postura de permiso preestablecido son los atacantes potenciales, porque quien mantiene el firewall no puede cubrir todos los agujeros, siempre esta en la modalidad de apaga fuegos y quizá este demasiado ocupado para advertir las actividades del atacante.

3.4.5 Filtrado de paquetes

El filtrado de paquetes es un mecanismo de seguridad de la red que funciona controlando que información puede fluir de y hacia una red. Aquí se proporcionara una introducción muy breve sobre los conceptos de alto nivel de una red IP (indispensable para entender el filtrado de paquetes).

Para transferir información a través de una red esta, debe dividirse en pequeñas partes, cada una de las cuales se envía en forma separada. Dividir la información en partes permite que varios sistemas compartan la red: cada uno envía fragmentos por turnos. En una red IP, los

pequeños elementos de información se llaman paquetes, toda la información transferida a través de las redes IP se hace en forma de paquetes.

El mecanismo básico que interconecta las redes IP se llama enrutador. Un enrutador puede ser una pieza dedicada de hardware sin algún otro propósito, o puede ser una pieza de software que se ejecuta en un sistema UNIX o en una computadora personal (con MS-DOS, Windows, Sistema Macintosh u otro) para uso general. Los paquetes que atraviesan una red de Internet (una red de redes) viajan de un enrutador a otro hasta que llegan a su destino.

3.4.5.1 El significado del filtrado de paquetes

El filtrado de paquetes permite controlar (permite o niega) la transferencia de información con base:

- La dirección de donde (supuestamente) proviene la información.
- La dirección a donde se dirige la información.
- Los protocolos de nivel sesión y aplicación que se emplean para transferir la información.

La mayoría de los sistemas para filtrado de paquetes no toman decisiones basándose en la propia información; tampoco basándose en el contenido.

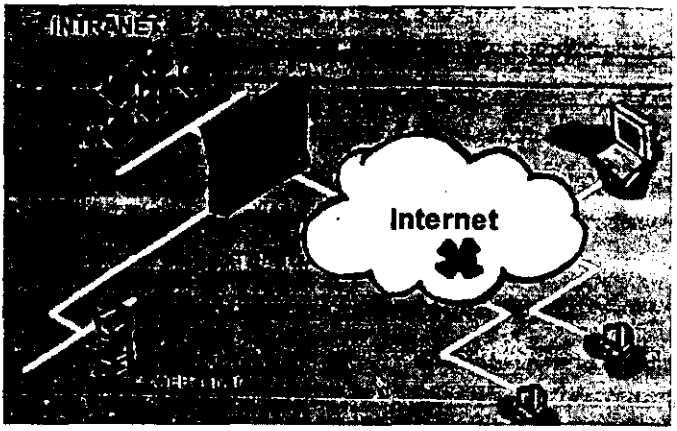
Ciertas protecciones pueden proporcionarse solo con los enrutadores con filtrado, y solo cuando se colocan en lugares específicos de la red.

3.4.5.2 Ventajas del filtrado de paquetes

Un enrutador de protección puede ayudan a proteger toda una red.

Una de las ventajas clave del filtrado de paquetes es que un solo enrutador con filtrado de paquetes colocado estratégicamente puede ayudar a proteger toda una red. Si solo hay un enrutador que conecta su sitio a Internet, logra una enorme ventaja en la seguridad de la red, sin importar el tamaño de su sitio al hacer el filtrado de paquetes en ese enrutador.
El filtrado de paquetes no necesita conocimiento o cooperación del usuario.

A diferencia de un proxy el filtrado de paquetes no requiere ningún programa o configuración especial por parte de las computadoras cliente, ni necesita algún adiestramiento especial o procedimiento por parte de los usuarios. Cuando un enrutador con filtrado de paquetes decide pasar un paquete, el enrutador es idéntico a un enrutador común. Lo ideal es que los usuarios no se percaten de que esta presente, a menos que intenten hacer algo prohibido por la política de filtrado del enrutador.



Esta transparencia quiere decir que el filtrado de paquetes puede hacerse sin la cooperación y con frecuencia sin el conocimiento de los usuarios. Lo importante es que puede hacer filtrado de paquetes sin que los usuarios tengan que aprender algo nuevo para que funcione y sin que usted. tenga que depender de que ellos hagan (o no) algo para que funcione.

Muchos enrutadores disponen de filtrado de paquetes.

Las capacidades de filtrado de paquetes están disponibles en muchos productos para enrutamiento de hardware y software, comerciales y/o disponibles gratuitamente en Internet. Muchos sitios ya cuentan con las capacidades de filtrado de paquetes en los enrutadores que usan.

Casi todos los enrutadores comerciales como Livingston, Enterprise y Cisco System, incluyen capacidades de filtrado de paquetes.

3.4.5.3 Desventajas del filtrado de paquetes

Aunque el filtrado de paquetes tiene muchas ventajas, también tiene muchas desventajas.

A pesar de la amplia disponibilidad para el filtrado de paquetes en varios productos de hardware y software el filtrado aún no es una arma perfecta, en mayor y menor grado esta capacidad en muchos de estos productos comparte limitaciones comunes:

- Las reglas para el filtrado tienden a ser difíciles de configurar. Aunque hay un rango de dificultad habitualmente va desde lo difícil hasta lo imposible.

- Una vez configuradas, las reglas para el filtrado tienden a ser difíciles de probar.

- Las capacidades para el filtrado de paquetes en muchos de los productos están incompletas, lo cual hace difícil o imposible la implantación de ciertos filtros sumamente deseables.

Como en cualquier otro aspecto, los programas para el filtrado de paquetes pueden tener problemas, generalmente son problemas de seguridad mas que errores de proxy. Un proxy que falla simplemente deja pasar la información, mientras que una falla de filtrado de paquetes podría permitir la entrada de paquetes que debieron rechazarse.

3.4.5.4 Convenciones para la regla de filtrado de paquetes

La sintaxis de filtrado especifica un número de bits significativos que se utilizan para compararlos con otras direcciones después de un carácter de separación (/). Por eso, 10.0.0.0/8 se acopla con cualquier dirección que empiece con 10; es equivalente a: 10.0.0.0 con una máscara de red 255.0.0.0, o lo mismo 10.0.0.0 con una máscara comodín de 0.255.255.255, o (sí fuera un nombre de archivo) 10.*.*.*.

El mecanismo exacto para establecer las reglas de filtrado de paquetes varía mucho de un producto a otro. Algunos productos (screend) permiten especificar un solo grupo de reglas que se aplican a todos los paquetes enrutados por el sistema. Otros (telbit, netblazer) permiten establecer reglas para interfaces específicas. Hay otros (como los productos Livingston y Cisco) que permiten especificar grupos de reglas y después de aplicarlos por nombres a interfaces especificas, por ejemplo, pueda definir un conjunto de regla compartido por uno número de interfaces diferentes, e instalar un grupo distinto las reglas que son únicas a una interfase dada.

3.4.5.5 Filtrado por dirección

La más simple, aunque no la más común, forma de filtrado de paquetes es el filtrado por dirección. Filtrar de este modo permite restringir el flujo de paquetes basándose en la dirección fuente y/o destino de los paquetes, sin tener que considerar que protocolos están involucrados. Tal tipo de filtrado puede emplearse para permitir que ciertos anfitriones externos conversen con ciertos anfitriones internos, o para evitar que un atacante cargue paquetes falsificados dentro de la red.

3.4.5.5 Filtrado por servicio

Detener paquetes falsificados de entrada es casi el único uso de los filtros basados en direcciones. Muchos otros usos de filtrados de paquetes implican el filtrado por servicio, lo que es un poco más complicado.

Desde el punto de vista de filtrado de paquetes como lucen los paquetes asociados a servicios específicos. Telnet permite que el usuario inicie sesión en otro sistema, como si estuviera en otra terminal, conectada directamente al ella, desde el punto de vista de filtrado de paquetes,

es representativo de muchos otros protocolos, como el smtp y nntp. Debemos observar el servicio de telnet saliente y entrante.

Un ejemplo más representativo de diferentes servicios de Internet para el filtrado de paquetes se especifican en la siguiente tabla:

Regla	Dirección	Dir. Fuente	Dir. Destino	Protocolo	Puerto Fuente	Puesto Destino	ACK	Acción
Spoof-1	entrada	interna	cualquiera	cualquiera	cualquiera	cualquiera	cualquiera	prohibir
Spoof-2	entrada	perímetro	cualquiera	cualquiera	cualquiera	cualquiera	cualquiera	prohibir
telnet-1	salida	interna	cualquiera	TCP	>1023	23	cualquiera	permitir
telnet-2	entrada	cualquiera	Interna	TCP	23	>1023	sí	permitir
ftp-1	salida	interna	cualquiera	TCP	>1023	21	cualquiera	permitir
ftp-2	entrada	cualquiera	Interna	TCP	21	>1023	sí	permitir
ftp-3	salida	interna	cualquiera	TCP	>1023	>1023	cualquiera	permitir
ftp-4	entrada	cualquiera	Interna	TCP	>1023	>1023	sí	permitir
ftp-5	salida	bastión	cualquiera	TCP	>1023	21	cualquiera	permitir
ftp-6	entrada	cualquiera	Bastión	TCP	21	>1023	sí	permitir
ftp-7	entrada	cualquiera	Bastión	TCP	20	6000-6003	cualquiera	permitir
ftp-8	entrada	cualquiera	Bastión	TCP	20	>1023	cualquiera	permitir
ftp-9	salida	bastión	cualquiera	TCP	>1023	20	sí	permitir
ftp-10	entrada	cualquiera	Bastión	TCP	>1023	21	cualquiera	permitir
ftp-11	salida	bastión	cualquiera	TCP	21	>1023	sí	permitir
ftp-12	salida	bastión	cualquiera	TCP	20	>1023	cualquiera	permitir
ftp-13	entrada	cualquiera	Bastión	TCP	>1023	20	sí	permitir
ftp-14	entrada	cualquiera	Bastión	TCP	>1023	>1023	cualquiera	permitir
ftp-15	salida	bastión	cualquiera	TCP	>1023	>1023	cualquiera	permitir
smtp-1	salida	bastión	cualquiera	TCP	>1023	25	cualquiera	permitir
smtp-2	entrada	cualquiera	Bastión	TCP	25	>1023	sí	permitir
smtp-3	entrada	cualquiera	Bastión	TCP	>1023	25	cualquiera	permitir
smtp-4	salida	bastión	cualquiera	TCP	25	>1023	sí	permitir
nntp-1	salida	Servidor NNT interno	Servidor NNTP de alimentación	TCP	>1023	119	cualquiera	permitir
nntp-2	entrada	Servidor NNT de alimentación	Servidor NNTP interno	TCP	119	>1023	sí	permitir
nntp-3	entrada	Servidor NNT de alimentación	Servidor NNTP interno	TCP	>1023	119	cualquiera	permitir
nntp-4	salida	Servidor NNT interno	Servidor NNTP de alimentación	TCP	119	>1023	sí	permitir
http-1	salida	Bastión	Cualquiera	TCP	>1023	cualquiera	cualquiera	permitir
http-2	entrada	cualquiera	Bastión	TCP	cualquiera	>1023	sí	permitir
http-3	entrada	cualquiera	Bastión	TCP	>1023	80	cualquiera	permitir
http-4	salida	Bastión	Cualquiera	TCP	80	>1023	sí	permitir
Dns-1	salida	Bastión	Cualquiera	UDP	53	53	a	permitir
Dns-2	entrada	cualquiera	Bastión	UDP	53	53	a	permitir
Dns-3	entrada	cualquiera	Bastión	UDP	cualquiera	53	a	permitir
Dns-4	salida	bastión	Cualquiera	UDP	53	cualquiera	a	permitir
Dns-5	salida	bastión	Cualquiera	TCP	>1023	53	cualquiera	permitir
Dns-6	entrada	cualquiera	Bastión	TCP	53	>1023	sí	permitir
Dns-7	entrada	cualquiera	Bastión	TCP	>1023	53	cualquiera	permitir
Dns-8	salida	bastión	cualquiera	TCP	53	>1023	sí	permitir
Default-1	salida	cualquiera	cualquiera	cualquiera	cualquiera	cualquiera	cualquiera	permitir
Default-2	entrada	cualquiera	cualquiera	cualquiera	cualquiera	cualquiera	cualquiera	Permitir

3.4.6 Implantación de filtrado de paquetes con ipfwadm

Conociendo como trabaja un firewall, en sus políticas de filtrado, ahora se hará la implantación de este en un sistema operativo Linux y con el comando de ipfwadm, para esto necesitamos saber que tipo de puertos y/o servicios necesitamos alojar en nuestro firewall, red interna y el exterior:

#Script que realiza un firewall para utilizar servicios tales como samba, #http, pop-3, "telnet", ftp, correo, DNS, ssh entre otros y negando todo lo #demás.

```
#Dando algunas variables

echo "Iniciando el Firewall"

FIRE_IP=`/sbin/ifconfig eth0 |grep 'inet addr'| awk '{print $2}'|sed -e
"s/addr\://"`
LOCAL="192.168.2.1"
FIRE_RED="192.168.2.0/24"
ALLIP="0.0.0.0/0"
HIPORTS="1024:65535"
#Negando las normas
echo "---------------- Negando las normas ------------------------------"
/sbin/ipfwadm -I -p deny
/sbin/ipfwadm -O -p deny
/sbin/ipfwadm -F -p deny

#Limpiando antiguos filtros
echo "---------------- Limpiando antiguos filtros ----------------------"
/sbin/ipfwadm -F -f
/sbin/ipfwadm -I -f
/sbin/ipfwadm -O -f

## Negando paquetes spoofed.
echo "------------------ Negando paquetes spoofed ----------------------"
/sbin/ipfwadm -I -a deny -V $FIRE_IP  -S $FIRE_RED  -D $ALLIP
/sbin/ipfwadm -I -a deny -V $FIRE_IP  -S $FIRE_IP -D $ALLIP

#Alojando trafico ilimitado entre la red local.
echo "------------ Alojando trafico ilimitado entre la red local -------"
/sbin/ipfwadm -I -a accept -V $LOCAL -S $ALLIP -D $ALLIP
/sbin/ipfwadm -O -a accept -V $LOCAL -S $ALLIP -D $ALLIP

echo "---------------- Internet:Firewall -------------------------------"
#
#Entrada de cualquier lado por puertos sin privilegios hacia los ptos. 137-#139,
#515(impresora),443(httpsd),110,80,53,25,22,21,20 y ping el servidor Firewall.
/sbin/ipfwadm -I -a accept -P icmp -S 132.248.59.2 -D $FIRE_IP
/sbin/ipfwadm -I -a accept -P tcp -S 132.248.59.60 515 -D $FIRE_IP 1020:65535
/sbin/ipfwadm -I -a accept -P tcp -S $ALLIP 1020:65535 -D $FIRE_IP 515 22
/sbin/ipfwadm -I -a accept -P tcp -S $ALLIP $HIPORTS -D $FIRE_IP 137 138 139 110
/sbin/ipfwadm -I -a accept -P tcp -S $ALLIP $HIPORTS -D $FIRE_IP 443 80 53 25 21
/sbin/ipfwadm -I -a accept -P tcp -S $ALLIP $HIPORTS -D $FIRE_IP 20
/sbin/ipfwadm -I -a accept -P udp -S $ALLIP $HIPORTS -D $FIRE_IP 53
#
#Preguntando a los servidores DNS root puerto 53
/sbin/ipfwadm -I -a accept -P tcp -S $ALLIP 53 -D $FIRE_IP 53
/sbin/ipfwadm -O -a accept -P tcp -S $FIRE_IP 53 -D $ALLIP 53
/sbin/ipfwadm -I -a accept -P udp -S $ALLIP 53 -D $FIRE_IP 53
/sbin/ipfwadm -O -a accept -P udp -S $FIRE_IP 53 -D $ALLIP 53
```

```
#Salida del servidor Firewall de los puertos 515, 137-139, 110, 80, 53, 25,
#22, 21 20 y ping  hacia Internet por puertos sin privilegios.
/sbin/ipfwadm -O -a accept -P icmp -S $FIRE_IP -D 132.248.59.2
/sbin/ipfwadm -O -a accept -P tcp -S $FIRE_IP 1020:65535 -D 132.248.59.60 515
/sbin/ipfwadm -O -a accept -P tcp -S $FIRE_IP 22 -D $ALLIP 1020:65535
/sbin/ipfwadm -O -a accept -P tcp -S $FIRE_IP 137 138 139 110 -D $ALLIP $HIPORTS
/sbin/ipfwadm -O -a accept -P tcp -S $FIRE_IP 443 80 53 25 21 20 -D $ALLIP $HIPORTS
/sbin/ipfwadm -O -a accept -P udp -S $FIRE_IP 53 -D $ALLIP $HIPORTS

echo "---------------- Firewall:Internet ------------------------------"
#
#Entrada de paquetes de los servidores de Internet de sus puertos 137-139,
#110, 80, 53, 25, 23, 22, 21, 20 y ping hacia los puertos sin privilegios del
#Servidor Firewall
/sbin/ipfwadm -I -a accept -P icmp -S $FIRE_IP -D 132.248.59.2
/sbin/ipfwadm -I -a accept -P tcp -S $ALLIP 137 138 139 110 -D $FIRE_IP $HIPORTS
/sbin/ipfwadm -I -a accept -P tcp -S $ALLIP 80 53 25 23 21 -D $FIRE_IP $HIPORTS
/sbin/ipfwadm -I -a accept -P tcp -S $ALLIP 20 -D $FIRE_IP $HIPORTS
/sbin/ipfwadm -I -a accept -P tcp -S $ALLIP 22 -D $FIRE_IP 1020:65535
/sbin/ipfwadm -I -a accept -P udp -S $ALLIP 53 -D $FIRE_IP $HIPORTS
#
#Salida de paquetes por los puertos sin privilegios del Servidor Firewall
#a servidores del Internet cuyos puertos son: 137-139, 80, 53, 25, 23,
#22, 21, 20 e icmp.
/sbin/ipfwadm -O -a accept -P icmp -S 132.248.59.2 -D $FIRE_IP
/sbin/ipfwadm -O -a accept -P tcp -S $FIRE_IP $HIPORTS -D $ALLIP 137 138 139 80
/sbin/ipfwadm -O -a accept -P tcp -S $FIRE_IP $HIPORTS -D $ALLIP 53 25 23 21 20
/sbin/ipfwadm -O -a accept -P tcp -S $FIRE_IP 1020:65535 -D $ALLIP 22
/sbin/ipfwadm -O -a accept -P udp -S $FIRE_IP $HIPORTS -D $ALLIP 53

echo "----------------------- Interna:Internet ------------------------------"
/sbin/ipfwadm -I -a accept -k -P tcp -S $ALLIP 110 53 25 21 80 23 -D $FIRE_RED
$HIPORTS
/sbin/ipfwadm -I -a accept -P tcp -S $ALLIP 20 -D $FIRE_RED $HIPORTS
/sbin/ipfwadm -I -a accept -P tcp -S $ALLIP 53 -D $FIRE_RED $HIPORTS
/sbin/ipfwadm -O -a accept -P tcp -S $FIRE_RED $HIPORTS -D $ALLIP 110 80 53 25 23
21 20
/sbin/ipfwadm -O -a accept -P udp -S $FIRE_RED $HIPORTS -D $ALLIP 53


#echo "------------------------Adelanto hacia el Internet -------------------"
/sbin/ipfwadm -F -a masquerade  -S $FIRE_RED -D $ALLIP
```

3.4.7 Implantación de filtrado de paquetes con ipchains

Poniendo otro ejemplo de cómo trabaja un firewall, en sus políticas de filtrado, ahora se hará la implantación en un sistema operativo Linux y con el comando de ipchains, para esto necesitamos saber que tipo de puertos y/o servicios necesitamos alojar en nuestro firewall, red interna y el exterior:

```
echo "Iniciando el firewall"
echo "--------------------- Asignando Variables --------------------"
FIRE_IP=`/sbin/ifconfig eth0 | grep 'inet addr' | awk '{print $2}'| sed -e
"s/addr\://"`
LOCAL="192.168.1.1"
FIRE_RED="192.168.1.0/24"
ALLIP="0.0.0.0/0"
```

```
HIPORTS="1024:65535"

echo "--------------------- Negando Normas --------------------"
/sbin/ipchains -P input DENY
/sbin/ipchains -P output DENY
/sbin/ipchains -P forward DENY

#Limpiando Antiguos Filtros
echo "------------------ Limpiando Filtros --------------------"
/sbin/ipchains -F input
/sbin/ipchains -F output
/sbin/ipchains -F forward

#Negando paquetes spoofed
echo "------------------ Negando paquetes Spoofed -------------"
#/sbin/ipchains -A input -j DENY -i eth0 -s $FIRED_RED -d $ALLIP
#/sbin/ipchains -A input -j DENY -i eth1 -s $FIRED_IP -d $ALLIP

#Alojando Trafico Ilimitado para la red Local paquetes spoofed
echo "------------- Alojando Trafico Ilimitado para la red Local -----------"
/sbin/ipchains -A input -j ACCEPT -V eth1 -s $ALLIP -d $ALLIP
/sbin/ipchains -A output -j ACCEPT -V eth1 -s $ALLIP -d $ALLIP

#Internet firewall
echo "------------- Internet:Firewall ---------------"
/sbin/ipchains -A input -j ACCEPT -p icmp -s 132.248.59.2 -d $FIRE_IP
/sbin/ipchains -A input -j ACCEPT -p tcp -s 132.248.59.60 515 -d $FIRE_IP
1020:65535
/sbin/ipchains -A input -j ACCEPT -p tcp -s $ALLIP 1020:65535 -d $FIRE_IP 515
/sbin/ipchains -A input -j ACCEPT -p tcp -s $ALLIP 1020:65535 -d $FIRE_IP 22
/sbin/ipchains -A input -j ACCEPT -p tcp -s $ALLIP $HIPORTS -d $FIRE_IP 137
/sbin/ipchains -A input -j ACCEPT -p tcp -s $ALLIP $HIPORTS -d $FIRE_IP 138
/sbin/ipchains -A input -j ACCEPT -p tcp -s $ALLIP $HIPORTS -d $FIRE_IP 139
/sbin/ipchains -A input -j ACCEPT -p tcp -s $ALLIP $HIPORTS -d $FIRE_IP 110
/sbin/ipchains -A input -j ACCEPT -p tcp -s $ALLIP $HIPORTS -d $FIRE_IP 80
/sbin/ipchains -A input -j ACCEPT -p tcp -s $ALLIP $HIPORTS -d $FIRE_IP 53
/sbin/ipchains -A input -j ACCEPT -p tcp -s $ALLIP $HIPORTS -d $FIRE_IP 25
/sbin/ipchains -A input -j ACCEPT -p tcp -s $ALLIP $HIPORTS -d $FIRE_IP 21
/sbin/ipchains -A input -j ACCEPT -p tcp -s $ALLIP $HIPORTS -d $FIRE_IP 20
/sbin/ipchains -A input -j ACCEPT -p udp -s $ALLIP $HIPORTS -d $FIRE_IP 53

#Preguntando a los servidores DNS root puerto 53
/sbin/ipchains -A input -j ACCEPT -p tcp -s $ALLIP 53 -d $FIRE_IP 53
/sbin/ipchains -A input -j ACCEPT -p udp -s $ALLIP 53 -d $FIRE_IP 53
/sbin/ipchains -A output -j ACCEPT -p tcp -s $FIRE_IP 53 -d $ALLIP 53
/sbin/ipchains -A output -j ACCEPT -p udp -s $FIRE_IP 53 -d $ALLIP 53

#Salida del servidor Firewall de los puertos 515,19-137, 110, 80, 53, 25, 22, 21
# 20 y ping hacia el Internet por puertos sin privilegios.
/sbin/ipchains -A input -j ACCEPT -p tcp -s $FIRE_IP 1020:65535 -d 132.248.59.60
515
/sbin/ipchains -A input -j ACCEPT -p tcp -s $FIRE_IP 22 -d $ALLIP 1020:65535
/sbin/ipchains -A input -j ACCEPT -p tcp -s $FIRE_IP 139 -d $ALLIP $HIPORTS
/sbin/ipchains -A input -j ACCEPT -p tcp -s $FIRE_IP 138 -d $ALLIP $HIPORTS
/sbin/ipchains -A input -j ACCEPT -p tcp -s $FIRE_IP 137 -d $ALLIP $HIPORTS
/sbin/ipchains -A input -j ACCEPT -p tcp -s $FIRE_IP 80 -d $ALLIP $HIPORTS
/sbin/ipchains -A input -j ACCEPT -p tcp -s $FIRE_IP 53 -d $ALLIP $HIPORTS
/sbin/ipchains -A input -j ACCEPT -p tcp -s $FIRE_IP 25 -d $ALLIP $HIPORTS
/sbin/ipchains -A input -j ACCEPT -p tcp -s $FIRE_IP 21 -d $ALLIP $HIPORTS
/sbin/ipchains -A input -j ACCEPT -p tcp -s $FIRE_IP 20 -d $ALLIP $HIPORTS
/sbin/ipchains -A input -j ACCEPT -p tcp -s $FIRE_IP 110 -d $ALLIP $HIPORTS
/sbin/ipchains -A input -j ACCEPT -p udp -s $FIRE_IP 53 -d $ALLIP $HIPORTS
```

```
echo "------------------ Firewall:Internet -----------------------------"
#
#Entrada de paquetes de los servidores de Internet de sus puertos 139-137,110,
#80, 53, 25, 23, 22, 21 y 20 hacia los puertos sin privilegios del servidor
#Firewall.
/sbin/ipchains -A input -j ACCEPT -p icmp -s $FIRE_IP -d 132.248.59.2
/sbin/ipchains -A input -j ACCEPT -p tcp -s $ALLIP 139 -d $FIRE_IP $HIPORTS
/sbin/ipchains -A input -j ACCEPT -p tcp -s $ALLIP 138 -d $FIRE_IP $HIPORTS
/sbin/ipchains -A input -j ACCEPT -p tcp -s $ALLIP 137 -d $FIRE_IP $HIPORTS
/sbin/ipchains -A input -j ACCEPT -p tcp -s $ALLIP 100 -d $FIRE_IP $HIPORTS
/sbin/ipchains -A input -j ACCEPT -p tcp -s $ALLIP 80 -d $FIRE_IP $HIPORTS
/sbin/ipchains -A input -j ACCEPT -p tcp -s $ALLIP 25 -d $FIRE_IP $HIPORTS
/sbin/ipchains -A input -j ACCEPT -p tcp -s $ALLIP 23 -d $FIRE_IP $HIPORTS
/sbin/ipchains -A input -j ACCEPT -p tcp -s $ALLIP 21 -d $FIRE_IP $HIPORTS
/sbin/ipchains -A input -j ACCEPT -p tcp -s $ALLIP 20 -d $FIRE_IP $HIPORTS
/sbin/ipchains -A input -j ACCEPT -p tcp -s $ALLIP 22 -d $FIRE_IP 1020:65535
/sbin/ipchains -A input -j ACCEPT -p udp -s $ALLIP 53 -d $FIRE_IP $HIPORTS
#
#Salida de paquetes por los puertos sin privilegios del Servidor Firewall
#a los servidores del Internet cuyos puertos son:139-137, 80, 53, 25, 23,
#22, 21 y 20.
/sbin/ipchains -A output -j ACCEPT -p icmp -s 132.248.59.2 -d $FIRE_IP
/sbin/ipchains -A output -j ACCEPT -p tcp -s $FIRE_IP $HIPORTS -d $ALLIP 139
/sbin/ipchains -A output -j ACCEPT -p tcp -s $FIRE_IP $HIPORTS -d $ALLIP 138
/sbin/ipchains -A output -j ACCEPT -p tcp -s $FIRE_IP $HIPORTS -d $ALLIP 137
/sbin/ipchains -A output -j ACCEPT -p tcp -s $FIRE_IP $HIPORTS -d $ALLIP 110
/sbin/ipchains -A output -j ACCEPT -p tcp -s $FIRE_IP $HIPORTS -d $ALLIP 80
/sbin/ipchains -A output -j ACCEPT -p tcp -s $FIRE_IP $HIPORTS -d $ALLIP 53
/sbin/ipchains -A output -j ACCEPT -p tcp -s $FIRE_IP $HIPORTS -d $ALLIP 25
/sbin/ipchains -A output -j ACCEPT -p tcp -s $FIRE_IP $HIPORTS -d $ALLIP 23
/sbin/ipchains -A output -j ACCEPT -p tcp -s $FIRE_IP $HIPORTS -d $ALLIP 21
/sbin/ipchains -A output -j ACCEPT -p tcp -s $FIRE_IP $HIPORTS -d $ALLIP 20
/sbin/ipchains -A output -j ACCEPT -p tcp -s $FIRE_IP 1020:65535 -d $ALLIP 22
/sbin/ipchains -A output -j ACCEPT -p udp -s $FIRE_IP $HIPORTS -d $ALLIP 53

echo "---------------------------- Interna:Internet ----------------------"
/sbin/ipchains -A input -j ACCEPT -p tcp -s $ALLIP 110 -d $FIRE_RED $HIPORTS
/sbin/ipchains -A input -j ACCEPT -p tcp -s $ALLIP 80 -d $FIRE_RED $HIPORTS
/sbin/ipchains -A input -j ACCEPT -p tcp -s $ALLIP 53 -d $FIRE_RED $HIPORTS
/sbin/ipchains -A input -j ACCEPT -p tcp -s $ALLIP 25 -d $FIRE_RED $HIPORTS
/sbin/ipchains -A input -j ACCEPT -p tcp -s $ALLIP 23 -d $FIRE_RED $HIPORTS
/sbin/ipchains -A input -j ACCEPT -p tcp -s $ALLIP 21 -d $FIRE_RED $HIPORTS
/sbin/ipchains -A input -j ACCEPT -p tcp -s $ALLIP 20 -d $FIRE_RED $HIPORTS
/sbin/ipchains -A input -j ACCEPT -p udp -s $ALLIP 53 -d $FIRE_RED $HIPORTS

echo "---------------------- Internet:Interna ---------------------------"
/sbin/ipchains -A output -j ACCEPT -p tcp -s $FIRE_RED $HIPORTS -d $ALLIP 110
/sbin/ipchains -A output -j ACCEPT -p tcp -s $FIRE_RED $HIPORTS -d $ALLIP 80
/sbin/ipchains -A output -j ACCEPT -p tcp -s $FIRE_RED $HIPORTS -d $ALLIP 53
/sbin/ipchains -A output -j ACCEPT -p tcp -s $FIRE_RED $HIPORTS -d $ALLIP 25
/sbin/ipchains -A output -j ACCEPT -p tcp -s $FIRE_RED $HIPORTS -d $ALLIP 23
/sbin/ipchains -A output -j ACCEPT -p tcp -s $FIRE_RED $HIPORTS -d $ALLIP 21
/sbin/ipchains -A output -j ACCEPT -p tcp -s $FIRE_RED $HIPORTS -d $ALLIP 20
/sbin/ipchains -A output -j ACCEPT -p udp -s $FIRE_RED $HIPORTS -d $ALLIP 53

echo "---------------------- Adelanto hacia el Internet -------------------"
/sbin/ipchains -A forward -j MASQ -s $FIRE_RED -d $ALLIP
echo "Ahh..................... todo acabo"
```

CAPÍTULO 4

4.1 Pruebas y resultados con el rendimiento del manejo del sistema de archivos

Se hizo la comprobación real de la eficiencia que tiene el sistema de archivos virtual (VFS) para la comunicación con otros sistemas de archivos, y se tiene como estadística que los sistemas de archivos reconocidos por Linux, puede interactuar sin ningún problema.

Esto coincidió con las tablas que se tienen en la página 55 y se llegó a resultados similares.

En cuanto a los sistemas de archivos en red (NFS) su rendimiento se ve afectado en forma exponencial, ya que este programa esta elaborado para que trabaje en forma eficiente en una red Lan (aproximadamente el 75% a 80%, como si se trabajará con un disco local).

Con esta aplicación podemos exportar discos de diferentes computadoras y se verán como si estuvieran en la computadora local y ellos tenemos que la información esta distribuido, se maneja con consistencia y en cualquier lugar que estemos podemos verla sin estar en la computadora que la almacena.

4.2 Resultados con la información de servicios de red

Esta aplicación se complementa con el punto anterior, ya que con esto además de poder tener la información en cualquier computadora y poder conectar desde cualquier computadora sin tener una cuenta real y exportar los archivos que necesitamos en común para el manejo de nuestros sistemas operativo.

Cabe señalar que el NIS (Servicio de Información en Red) ya ha sido mejorado con otra aplicación llamada NIS+, el concepto es el mismo pero tiene autentificación de las computadoras clientes y cifrado de los datos de la red, ya que en el NIS cualquiera puede ser cliente si conoce el dominio de este servicio y puede traer los datos de una manera visual y la otra desventaja es que se pueden ver los datos que viajan en claro y cualquier aplicación de tipo sniffer puede ver lo que llega a ese puerto.

4.3 Pruebas y resultados con la aplicación SAMBA

Esta aplicación ha tenido un gran fruto en el laboratorio de computadoras y programación del Edificio Valdés Vallejo, planta baja, Facultad de Ingeniería, UNAM ya que esta configuración ha hecho que la información de cuentas de alumnos, manejo de impresoras y aplicaciones comunes en un sistema operativo como Windows NT radique en un servidor Linux, además la computadora trabaja con un buen performance atiende a 100 computadoras conectadas y utilizan algún recurso de este servidor y es totalmente transparente para el usuario y con las ventajas de los permisos y la filosofía de UNIX como:

- Permisos de archivos, directorios.
- Cuota sobre las cuentas de los usuarios en Windows (configuración de cuota).
- Manejo y cuota en las impresoras.

Además de que los archivos, que ciertos usuarios no necesitan conocer, se pueden ocultar y controlar la sesión desde el servidor Linux.

Esto se debe al protocolo anexado a Linux para comunicación con Windows, que en realidad es el protocolo nativo de Windows y con ello nos lleva a que la manipulación de usuarios Linux, no se den cuenta del sistema operativo que están usando.

4.4 Pruebas y resultados con la aplicación IP-Masquerade

Esta aplicación se tuvo que utilizar por necesidad pero ha funcionado tanto que actualmente el dominio fi-b.unam.mx cuenta con más de 400 computadoras y solamente se tienen 254 direcciones ip y usando la aplicación tenemos direcciones ip libres y todas las computadoras salen a Internet, esta aplicación se instalo en varios laboratorios:

- Laboratorio de Interfaces Inteligentes, Laboratorio de Investigación para el Desarrollo Académico (LINDA), Laboratorio de Computadoras y Programación, Planta Baja, Edificio Valdés Vallejo, División de Ingeniería Eléctrica, Facultad de Ingeniería.
- Laboratorio de Computación, Segundo Piso, Edificio Valdés Vallejo, División de Ingeniería Eléctrica, Facultad de Ingeniería.
- Laboratorio de Informática, Segundo Piso, División de Estudios de Posgrado, Facultad de Ingeniería.

La velocidad de transmisión usando un gateway no afecta demasiado en el retardo de datos, el usuario no se da cuenta de esta cantidad de perdida, hablando de milésimas de segundo.

4.5 Resultados con la aplicación IPFWADM/IPCHAINS (firewalls)

Este tipo de aplicaciones sirve para conocer quien posee un mejor su ruteo en cuanto a los paquetes aceptados y la velocidad para atenderlos, los dos lo hacen usando el kernel, esto ayuda a que la velocidad para ver los paquetes sea muy rápida, la única desventaja radican el programa **ipfwadm** porque que tiene muchas inconsistencias y banderas en cuanto a los paquetes TCP/IP que no se tienen considerados y puede ser un punto para poder evadir el firewall.

CONCLUSIONES

En el presente trabajo se implantaron las diversas aplicaciones que ofrece Linux para la comunicación entre computadoras y así resolver problemas comunes, con ello se podrá publicar este trabajo en la Internet como manual de ayuda a los que deseen usarlo, en cuanto a investigación y desarrollo de estos capítulos, teniendo más acceso a este sistema operativo como sus aplicaciones y problemas resueltos.

Aunque este proyecto fue realizado a nivel experimental dando algunos frutos, posteriormente fue llevado a la práctica publicándolo en Internet, esto ha llevado a su consulta en varias instituciones dentro y fuera del país como en:

- España
- Francia
- Venezuela
- Argentina
- Colombia.

Se ha comprobado las diferentes aplicaciones y sus resultados.

En la realización del servidor de sistemas de archivos compartidos mediante la aplicación SAMBA, se analizó el funcionamiento de este protocolo entre computadoras con plataforma Windows y la manera de implantarlo en Linux y posteriormente se realizó en UNIX que es de manera similar y se puede modificar de manera manual las opciones de este protocolo y ver los efectos que tienen al momento de compartir recursos.

Una de las maneras de realizar una red de computadoras en cuanto a simplicidad tanto en la realización y económico es por medio de la comunicación serial, donde se verifica y fortalece la existencia de los protocolos SLIP, PPP, entre otros para realizar estas redes de poco alcance pero muy útiles al momento de compartir recursos.

En el análisis de cómo trabaja un filtrado de paquetes, es mas que estudiar el protocolo TCP/IP, se debe conocer como trabaja cada protocolo de la capa de aplicación para hacer un filtrado individual y no grupal además de que sea optimo y no cuando se empezó este documento que funcionaba de manera grupal, ya que cada aplicación que utiliza este protocolo, se debe de estudiar la manera como trabaja y poder manipular de manera optima este protocolo.

En la elaboración del gateway se construyó una computadora con doble acceso, aquí se puso todo el conocimiento para el entendimiento de todo lo anterior y su posible solución.

Una gran ventaja de estas prácticas es la curva de aprendizaje, el lector aprenderá más rápido en poco tiempo. Se ha demostrado una parte de esta curva en donde se impartieron en 5 semestres diferentes cursos con estas prácticas en la Facultad de Ingeniería y han desarrollado los alumnos estás prácticas, teniendo beneficios en ellas.

Al adquirir los conocimientos del sistema operativo Linux y si en un futuro el alumno manejará otro sistema operativo UNIX, el alumno lo trabajará y usará de manera semejante al sistema operativo Linux y lo dominará sin ningún problema.

En cuanto a esta tesis siempre será una documentación de mejora continua, por que se podrá enriquecer el documento, con otras propuestas o mejorándose la que están escritas y probadas, dado que el sistema operativo Linux está renovación constante y está documentación para estar al día debe de tomar esta filosofía y así estar a la par de todas las distribuciones actuales.

La experiencia de contribuir a la Facultad de Ingeniería a soluciones tendiendo pocos recursos asignados y con ello tener una infraestructura similar a compañías que tengan software propietario, que al momento de comparar las dos soluciones de estas tecnologías son casi iguales en su desempeño y muy diferente en costo.

Así se da una solución de controlar una red interna como de impresión y usuarios con el sistema operativo Linux y sus aplicaciones, aunque sea heterogénea los sistemas operativos y estos cuentan con la conexión a Internet para uno o varios servicios, los demás servicios serán filtrados o negar su acceso; también la autentificación de usuarios desde un servidor, siendo este un sistema operativo diferente al que lo esta solicitando como cliente.

Se encontró que existe muy poca información detallada sobre el conocimiento y solución de estos problemas en cuanto a Linux, así que esta tesis puede ayudar a simplificar el diseño de las aplicaciones que hagan uso del sistema operativo Linux. Por otro lado, la aportación que este trabajo hace a la Facultad de Ingeniería a una posible aplicación y creación de un taller de redes de computadoras, basándose en la realización de experimentos de las características más representativas de las redes de computadoras, con el objetivo de ver en forma tangible los conceptos del Modelo de Referencia OSI (Open System Interconnection), el cual comúnmente es tratado de forma abstracta en su enseñanza y la comparación del Protocolo TCP/IP, con esto se llevará a que la materia ya no sea teórica, sino práctica.

APENDICES

Apéndice A Diferencias entre ipchains e ipfwadm

Algunos de estos cambios son un resultado de modificaciones en el kernel, y como resultado es la diferencia entre ipchains e ipfwadm.

1. Muchos argumentos han sido remapeados: Las mayúsculas ahora indican un comando, y las minúsculas una opción.

2. Se soportan cadenas arbitrarias, incluso cadenas de construcción que tienen nombres completos en lugar de flags (ej. `input ' en lugar de ` -I ').

3. La opción ` -k ' ha desaparecido: use `! -y '.

4. La opción ` -b ' realmente inserta/adiciona/borra dos reglas, en lugar de una sola regla 'bidireccional'.

5. La opción ` -b ' puede pasarse a ` -C ' para hacer dos chequeos (uno en cada dirección).

6. La opción ` -x ' para ` -l ' se ha reemplazado por ` -v '.

7. Múltiples puertos de origen y destino no son soportados más. Pudiendo negar un rango de puertos que puede hacer mas que eso.

8. Las interfaces sólo pueden ser especificadas por nombre (no por dirección). De cualquier forma la vieja semántica fue cambiada en el kernel 2.1.

9. Los fragmentos son examinados, no se permiten que crucen automáticamente.

10. Las cadenas de accounting explícitas se han hecho afuera.

11. Los protocolos arbitrarios por encima de IP pueden probarse.

12. El viejo comportamiento de los emparejamientos de SYN y ACK (previamente ignorados para paquetes NO-TCP) ha cambiado. La opción de SYN no es válida para las reglas que no especificas de TCP.

13. Los contadores son ahora de 64 bits en computadoras de 32 bits, no de 32 bits.

14. Se soportan opciones inversas ahora.

15. Códigos ICMP son ahora soportados.

16. Interfaces Wildcard son soportadas ahora.

17. Las manipulaciones de TOS están ahora revisadas de sanidad: el viejo código de kernel lo detendría silenciosamente (ilegalmente) manipulando el bit TOS "Must be Zero"; ipchains ahora retorna un error si lo intenta, tal como para otros casos ilegales.

Tabla de referencia rápida

[Principalmente, los argumentos de los comandos son en MAYUSCULAS, y los argumentos en minúsculas]

Una cosa para notar, el enmascaramiento se especifica por ` -j MASQ'; es completamente diferente de ` -j ACCEPT', y no trabaja como efecto lateral, diferente a lo que hace ipfwadm

ipfwadm	ipchains	Notas
-A [both]	-N acct -I 1 input -j acct -I 1 output -j acct acct	Crea una cadena 'acct' y tiene paquetes salientes y entrantes cruzándola.
-A in	input	Una regla sin objetivo
-A out	output	Una regla sin objetivo
-F	forward	Use esto como [cadena]
-I	input	Use esto como [cadena]
-O	output	Use esto como [cadena]
-M -l	-M -L	
-M -s	-M -S	
-a policy	-A [chain] -j POLICY	(ver -r y -m).
-d policy	-D [chain] -j POLICY	(ver -r y -m).
-i policy	-I 1 [chain] -j POLICY	(ver -r y -m).
-l	-L	
-z	-Z	
-f	-F	
-p	-P	
-c	-C	
-P	-p	
-S	-s	Solo toma un puerto o un rango, no múltiples.
-D	-d	Solo toma un puerto o un rango, no múltiples.
-V		Use -i [nombre]
-W	-i	
-b	-b	Ahora hace 2 reglas.
-e	-v	
-k	! -y	No trabaja a menos que se especifique -p tcp.
-m	-j MASQ	
-n	-n	

-o	-l	
-r [redirpt]	-j REDIRECT [redirpt]	
-t	-t	
-v	-v	
-x	-x	
-y	-y	No trabaja a menos que se especifique -p tcp

Ejemplos de comandos ipfwadm trasladados

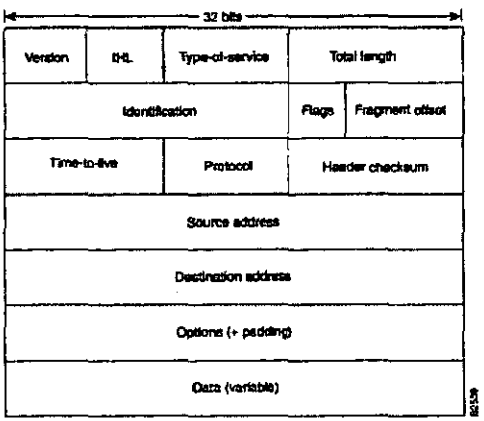
```
Antiguo comando: ipfwadm -F -p deny
Nuevo comando: ipchains -P forward DENY

Antiguo comando: ipfwadm -F -a m -S 192.168.0.0/24 -D 0.0.0.0/0
Nuevo comando: ipchains -A forward -j MASQ -s 192.168.0.0/24 -d 0.0.0.0/0

Antiguo comando: ipfwadm -I -a accept -V 10.1.2.1 -S 10.0.0.0/8 -D
0.0.0.0/0
Nuevo comando: ipchains -A input -j ACCEPT -i eth0 -s 10.0.0.0/8 -d
0.0.0.0/0
```

No hay ningún equivalente para especificar interfaces por dirección: se usa el nombre de la interfase. En esta computadora, 10.1.2.1 corresponde a eth0.

Apéndice B Cabeceras IP



Version: 4 bits.

El campo de version indica la forma de la cabecera de Internet.

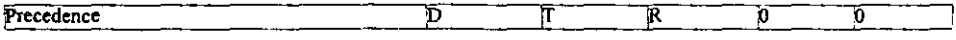
IHL: 4 bits.

La longitud de cabecera de Internet es la longitud de la cabecera en 32 bits de la palabra. El mínimo valor de una cabecera correcta es 5.

Tipo de Servicio: 8 bits.

El tipo de servicio indica la calidad del servicio deseado.

Bits 0-2: Precedente.
Bit 3: 0=Retraso Normal, 1= Retraso Bajo.
Bit 4: 0=Rendimiento Normal, 1= Rendimiento Alto.
Bit 5: 0=Integridad Normal, 1= Integridad Alta.
Bits 6-7: Reservar para uso futuro.



Precendente
111.-Control de Red
110.-Control de InternetWork
101.-Critic/ECP
100.-FlashOverride
011.-Flash
010.-Inmediate
001.-Prioridad
000.-Rutina

Total de Longitud: 16 bits.

La longitud total es la longitud del datagrama, están en bytes, incluyendo la cabecera de Internet y datos.

Identificación: 16 bits.

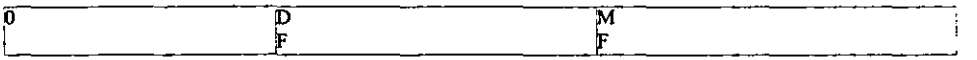
Un valor para identificar y asignar por donde se manda y ensamblados fragmentos del datagrama.

Banderas: 3 bits.

Varias banderas de control. El bit de la bandera son explicados a continuación:

Bit 0: Reservado, debe ser cero.
Bit 1: (DF) 0= Esta fragmentado, 1=No fragmentado.

Bit 2: (MF) 0=Ultimo fragmento, 1=Mas fragmentos.



Offset del Fragmento: 13 bits.

Este campo indica cuando en el datagrama esta fragmentado. El offset del fragmento esta medido en unidades de 8 bytes (64 bits). El primer fragmento tiene un offset de cero.

Tiempo de vida: 8 bits.

Este campo indica el máximo tiempo del datagrama es aloja en el sistema del Internet.

Protocolo: 8 bits.

Este campo indica el protocolo de la capa de transporte esta la porción del dato de este datagrama pasado. El valor de varios protocolos son específicos en el RFC "asignación de números".

Cabeceras Checksum: 16 bits.

Un checksum en la cabecera únicamente. Alguna cabecera cambia (ejemplo el Tiempo de Vida), esta es verificada en cada punto. esto la cabecera de internet es procesada. El algoritmo del checksum es el siguiente:

El campo del checksum es el 16 bit complemento de los primeros complementos suma el total los 16 bits de la palabra de la cabecera. La propuesta del computo del checksum, el valor del checksum de campo es cero.

Dirección Fuente: 32 bits.

La dirección IP origen

Dirección Destino:

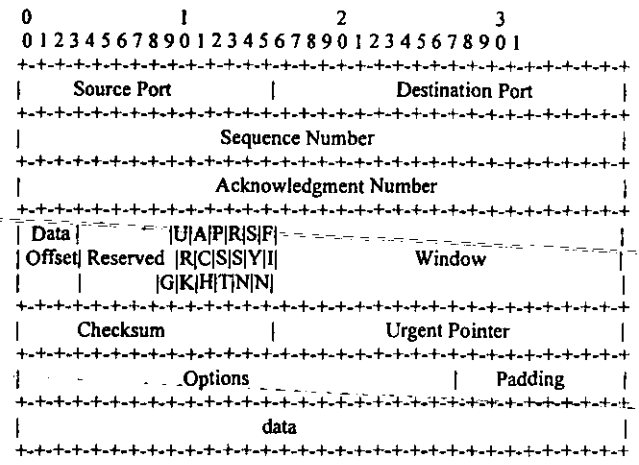
La dirección ip destino.

Opciones: Variable

La opción puede o pueden no apegarse en datagramas, pero esto debe ser implantado por todos los módulos IP (host y gateways).

Cabeceras de segmentos TCP

Esta descripción esta tomada de las páginas RFC 15 al 17 del RFC 793.



Puerto origen: 16 bits.

El número de puerto origen.

Puerto Destino: 16bits.

El número de puerto destino.

Número de secuencia: 32 bits.

La secuencia de números de los primeros datos de 8 bits (byte) en este segmento (excepto cuando esta presente SYN). Si SYN esta presente la secuencia de números es el número inicial secuente (ISN) y el primer dato de 8 bits es ISN+1.

Número Acknowledgment: 32 bits.

Si el control de ACK esta puesto, este campo contiene el valor de la próxima secuencia de números a mandar del segmento excepto a recibir. Una conexión establecida esta siempre esta puesta.

Offset del Dato: 4 bits.

El número de 32 bits de palabra en la cabecera de TCP. Esto indica cuando el dato comienza. La cabecera TCP (donde incluyen opciones) es un número entero de 32 bits de largo.

Reservado: 6 bits.

Reservado para uso futuro. Debe ser cero.

Control de bits: 6 bits-singles valores (de izquierda a derecha):

URG: Campo significativo de Urgent Pointer
ACK: Campo significativo de Acknowledgment
PSH: función Push.
RST: Reset la conexión.
SYN: Número de secuencia sincronizados (Synchronize).
FIN: No mas datos a mandar.

Window: 16 bits.

El número de datos en bytes al mandar este segmento esta aceptado.

Checksum: 16 bits.

El campo de checksum es el complemento de los primeros 16 bits de los primeros complementos de toda la suma de las 16 palabras en la cabecera y texto.

Urgent Pointer: 16 bits

Este campo contiene el valor actual del urgent pointer como un offset positivo del número de secuencia en este segmento. El urgent pointer de la secuencia de números de los octetos siguientes los datos de urgent. Este campo únicamente es interpretado en el segmento con el control de bit URG puesto.

Opciones: Variable

Opciones pueden ocupar más espacio como el final de la cabecera TCP y son un múltiplo de 8 bits de longitud.

Cabeceras de los Mensajes de problemas de parámetros ICMP.

Esta descripción fue tomada de las páginas 8 y 9 del RFC 792, Internet Control Messages Protocol.

Tipo: 12

Code: 0= Punto que indica el error.

Checksum:

El campo de checksum es el complemento de los primeros 16 bits de los primeros complementos de toda la suma del mensaje ICMP inicia con el tipo ICMP. El computo hace el checksum y el campo de checksum debe ser cero.

Punto: Si code=0, identifica el byte cuando un error fue detectado.

Cabecera de Internet + 64 bits de Datagrama de Datos.

La cabecera de Internet mas los primeros 64 bits del datagrama entonces responde el error.

GLOSARIO

ARP. Protocolo de resolución de direcciones, define el método para usar un mapeo de la dirección IP de 32 bits a una dirección Ethernet de 48 bits.

Backbone. El mecanismo de conectividad primaria de una red. Todos los sistemas tienen conectividad en el backbone puede tener conectividad con otra.

Cracker. Un cracker es alguno que rompe la seguridad de los sistemas de computadoras o software con malas intenciones.

Cubos. El cubo sirve como un punto de conexión para cables coaxiales y puede ser pasivo o activo. Un cubo pasivo tiene, por lo general cuatro conectores. Un cubo activo tiene casi siempre, ocho puertos y amplifica o revela la señal.

Daemon. Programa del sistema que esta ejecutando en background y que realiza tareas periódicas relacionadas con la administración del sistema. A diferencia de muchos comandos que se ejecutan y se acaban, un daemon realiza su trabajo y se espera más.

Datagrama. Se usa de manera indistinta con paquete de datos o mensaje de red para identificar una unidad de información que se intercambia.

DHCP. Protocolo de configuración dinámica de hosts, automatiza la asignación de direcciones IP en una organización de red.

DNS. Significa "servicio (ó sistema) de nomenclatura (ó nombres) de dominio. Es un servicio que proporciona una ó más computadoras de la red para ayudar a localizar una ruta hacia un nodo deseado. Esto ahorra a cada sistema de la red el mantener una lista de todos los sistema con los que quiere comunicarse, lo usan las compuertas de correo.

Gateway (Compuerta). Se usa la compuerta cuando se necesita conectar tipos de redes que no son similares. La compuerta ejecuta la conversión de protocolos requerida para que las redes puedan comunicarse.

Gosip. Significa "perfil de interconexión de sistemas abiertos gubernamentales" . Es una colección de protocolos OSI, usados en las redes de computadoras y en los proyectos del Gobierno de los Estados Unidos.

GUI. Interfaz gráfica de usuario.

FTAM. Significa "transferencia, acceso y manejo de archivos".Es un protocolo de transferencia y administración de archivos especificados por OSI.

FTP. Significa "protocolo de transferencia de archivos". Es un protocolo de transferencia y de archivos entre sistemas, con velocidad muy considerable.

Hacker. Alguien interesado en sistemas operativos, software, seguridad y la Internet en general. También un programador.

HTTP. Protocolo de transferencia de hipertexto, es usado en el World Wide Web (WWW) para intercambiar texto, imágenes, sonido y otra información multimedia vía una interfaz gráfica de usuario (GUI)

Host. Una computadora con una dirección de hardware permanente, especialmente en una red TCP/IP.

ICMP. Protocolo de mensajes de control de Internet, comunica los mensajes de error y otros de control entre datagramas IP.

IP. Significa "protocolo de internet". Es un protocolo responsable de transportar datagramas por Internet.

NFS. Significa "sistemas de archivos en red". Es un sistema de disco virtual en red que permite a una computadora cliente montar sistemas y directorios de archivos remotos. Lo desarrolló en su origen SUN MICROSYSTEM.

NIC. Centro de información de red. Es el responsable de administrar Internet, las direcciones TCP/IP y los nombres de red.

Nodo. Es una computadora en red.

OSI. Significa "interconexión de sistemas abiertos" . Es el modelo estándar de ISO para la definición de comunicaciones de datos.

POP-3. Protocolo de oficina postal, versión 3, aloja usuarios a leer correo electrónico sobre una red desde un servidor central.

PPP. Protocolo punto a punto, transmite datagramas sobre ligas seriales punto a punto.

Protocolo. Un protocolo es una serie de reglas para intercambiar datos entre 2 entidades, estas reglas son:

Sintaxis: Formatos de datos y códigos.

Semántica: Control de la información y manejo del error.
Tiempo: Velocidad y secuencia.

Proxy. Programa que trata con servidores externos en nombre de los clientes internos, son programas representantes.

Puentes. Se usa el puente cuando se necesita conectar 2 tipos de redes similares entre ellos.

RARP. Protocolo de reversa de resolución de direcciones, es lo inverso de ARP. Hace el mapeo de la dirección Ethernet de 48 bits a 32 bits de la dirección IP.

Repetidores. Los repetidores amplifican o regeneran la señal sobre la red, para que se puedan extender los límites de distancia normales de cableado de la red.

RFC. Significa "petición de comentarios" . Es la documentación que mantiene el NIC en cuanto a protocolos de Internet, direccionamiento, configuración y otros temas de Internet relacionados.

RIP. Es el protocolo que especifica la información de la ruta entre gateways. Se utiliza para intercambiar información entre ruteadores.

rlogin. Un servicio de los sistemas UNIX en cual aloja usuarios de una computadora a otra computadora UNIX.

RMON. Significa "monitor remoto". Es un monitor de red remoto que permite recolectar información acerca del tráfico en la red.

RPC- Es llamada remota de procedimiento. Permite ejecutar procedimientos en un servidor.

Ruteadores. Los ruteadores se utilizan en redes grandes y complejas a través de diversas rutas de acceso por donde las señales de la red pueden viajar al mismo destino. El ruteador determina y envía la señal por la ruta de acceso más fácil.

SLIP. Línea serial IP, encapsula los datagramas IP en líneas seriales.

SMTP. Significa "protocolo simple de transferencia de correo". Se usa para transferir correo electrónico entre sistemas.

Sniffer. Programa que captura datagramas sobre la red. Esto se puede usar legítimamente (por un ingeniero de sistemas al diagnosticar problemas en la red) o ilegítimamente (por un cracker mirando los usuarios y contraseñas).

SNMP. Es el protocolo de administración simple de red. Es un protocolo usado para administrar dispositivos de red remotos y recolectar información de estos relativo a su configuración, errores y alarmas.

Switch. Un dispositivo multipuerto el cual provee de la dinámica lógica de conexión y desconexión entre alguno de los dos segmentos de cable sin intervención de l operador. El switch es un dispositivo de alta velocidad.

TCP. Significa "protocolo de control de transmisión". Es el protocolo entre un par de aplicaciones, responsable de transmitir datos de manera confiable y con una orientación a conexiones.

Telnet. Es el protocolo usado para establecer conexiones de terminales remotas.

UDP. Es el usuario de protocolo datagrama; no tiene conexión y se emplea para transferir datos entre agentes.

VT. Quiere decir terminal virtual. Es un método para usar Telnet, a fin de registrarse en sistemas remotospor medio de la red..

REFERENCIAS

1) Williams S., "*Comunicaciones y Redes de Computadoras*", editorial Prentice Hall, quinta edición, España, 1997.

2) Brent C. y Elizabeth Z,"*Construya Firewalls para Intenet*", editorial O'Reilly, segunda edición, México, 1998.

3) Anonynous,"*Maximum Linux Security*",editorial sams, segunda edición, USA, 1998.

4) Cheswick, B. , "Firewalls and Internet Security", editorial addison wesley, doceava edición, USA, 1999

5) Paul G. , " *Linux Network Toolkit*", editorial IDG Books, primera edición, USA, 1998

6) John B. , "Pow*er Programming with RPC*", editorial O'Reilly, primera edición, USA, 1993

7) Garfinkel y S. ,"*Seguridad Práctica en Unix e Internet*", editorial Mc Graw Hill, segunda edición, USA, 2000.

8) New Rider's. "*Firewall's y la Seguridad en Internet*", editorial Prentice Hall, segunda edición, México 1997

9) Uyless B., "*Redes de Computadoras*", editorial Addison Wesley, segunda edición, USA, 1995.

10) Olaf K. , "*Linux Network Administration*", libro no comercial, 1995.

11) Craig H., "*TCPI/IP Network Administration*", editorial O'Reilly, segunda edición, USA, 1995.

12) Matt W. ,"*Linux Installation and Getting Started*", libro no comercial, 1995.

13) Andrew S. Tanenbaum. "*Redes de Computadoras*", editorial Prentice Hall, tercera edición, México, 1997.

14)Anonynous. "*Maximum Security*", editorial SAMS, primera edición, USA, 2000.

15)Cricket L. , Jerry P. , Russ J. , Bryan B. y Adrian N., "*Administración de Servicios de Información en Internet*", editorial Mc Graw Hill, primera edición, México, 1995.

16) Peek, O'Reilly and Loukides, "*Unix Power Tools*", editorial O'Reilly, segunda edición, USA, 1998

17) Francisco Manuel Marquez G. , *"Unix Programación Avanzada"*, Editorial Addison-Wesley, primera edición, USA, 1993

18) Manejo y administración de memoria
http://www.redhat.com:8080/HyperNews /get/memory/80386mm.html

19) Documentación de redhat
http://www.redhat.com/

20) Documentación de firewall e ipfwadm
http://www.xos.nl/Linux/ipfwadm

21) Introducción al ext2fs
http://www.redhat.com:8080/HyperNews /get/memory/ext2intro.html