

REDES BAJO LINUX

ÍNDICE DE ILUSTRACIONES

Ilustración 3-1 (Modelo centralizado).....	19
Ilustración 3-2 (Adaptador multipuerto)	20
Ilustración 3-3 (Modelo distribuido).....	21
Ilustración 3-4 (Modelo Cliente/Servidor)	23
Ilustración 3-5 (Tarjeta Fast Ethernet).....	24
Ilustración 3-6 (Cable coaxil fino).....	25
Ilustración 3-7 (Conector tipo BNC).....	25
Ilustración 4-1 (Módem tipo tarjeta)	34
Ilustración 4-2 (Módem tipo externo)	34
Ilustración 6-1 (Topología en bus)	59
Ilustración 6-2 (Topología en estrella).....	60
Ilustración 6-3 (Topología en anillo).....	61
Ilustración 6-4 (Modelo de red interna a la facultad)	62
Ilustración 6-5 (Modelo de intranet Biblioteca 2000).....	64

ÍNDICE GENERAL

1. INTRODUCCIÓN	4
2. BREVE HISTORIA DE LINUX	7
2.1 CARACTERÍSTICAS DEL SISTEMA.....	9
2.1.1 <i>Multitarea</i>	11
2.1.2 <i>Multiusuario</i>	12
2.2 DIFERENCIAS ENTRE LINUX Y OTROS SISTEMAS OPERATIVOS	13
3. INTRODUCCIÓN A LAS REDES	14
3.1 HISTORIA	14
3.2 REDES.....	15
3.3 LA IMPORTANCIA DE UNA ADMINISTRACIÓN ADECUADA.....	16
3.3.1 <i>Acerca de raíces, sombreros y la sensación de poder</i>	16
3.3.2 <i>La cuenta root</i>	17
3.4 COMPRENSIÓN DE LOS SISTEMAS DE PROCESO CENTRALIZADO	18
3.4.1 <i>Elementos del modelo de procesamiento centralizado</i>	19
3.5 COMPRENSIÓN DE LOS SISTEMAS DE PROCESAMIENTO DISTRIBUIDO	21
3.5.1 <i>Los elementos del modelo de procesamiento distribuido</i>	21
3.5.2 <i>Revisión de topologías</i>	22
3.6 MODELO CLIENTE/SERVIDOR.....	23
3.7 ETHERNETS	24
3.7.1 <i>Tarjetas Ethernet</i>	26
3.8 OTROS TIPOS DE HARDWARE.....	27
3.9 REDES CON LINUX	28
3.9.1 <i>Diferentes etapas de desarrollo</i>	29
3.10 EL PROTOCOLO IP (INTERNET PROTOCOL).....	30
3.10.1 <i>Problemas con números IP en Internet</i>	31
4. IP EN LÍNEAS SERIE, SLIP Y PPP	33
4.1 CONFIGURANDO EL SOFTWARE SERIE.....	33
4.1.1 <i>Software de comunicaciones vía módem</i>	33
4.2 INTRODUCCIÓN A DISPOSITIVOS SERIE	35
4.3 ACCEDIENDO A LOS DISPOSITIVOS SERIE	36
4.4 HARDWARE SERIE	37
5. EL PROTOCOLO DE CONTROL DE TRANSMISIÓN, TCP.....	40
5.1 HARDWARE REQUERIDO.....	42

5.2 CONFIGURACIÓN DE TCP/IP	43
5.2.1 <i>La configuración de red</i>	43
5.3 INTRODUCCIÓN AL DNS	45
5.3.1 <i>Búsquedas de nombres con DNS</i>	47
5.3.2 <i>Servidores de nombres</i>	48
5.3.3 <i>La base de datos DNS</i>	49
5.4 LOS FICHEROS DE INICIO RC PARA TRABAJO EN REDES	50
5.4.1 <i>/etc/hosts</i>	53
5.4.2 <i>/etc/networks</i>	54
5.4.3 <i>/etc/host.conf</i>	54
5.4.4 <i>/etc/resolv.conf</i>	54
5.4.5 <i>Ajuste del nombre de su host</i>	55
5.4.6 <i>Problemas con la configuración</i>	55
5.5 LA LIBRERÍA DE SOCKETS.....	56
6. APÉNDICE A. EL PROYECTO.....	57
6.1 <i>¿QUÉ ES INTRANET?</i>	63
6.2 <i>INTRANET VS. INTERNET</i>	63
6.3 <i>EL PAPEL DE LINUX EN INTRANET</i>	64
6.4 <i>USUARIOS Y FORMACIÓN</i>	65
7. APÉNDICE B. LA LICENCIA GENERAL PÚBLICA. GNU.....	67
7.1 <i>PREÁMBULO</i>	67
7.2 <i>TÉRMINOS Y CONDICIONES DE LA LICENCIA GENERAL PÚBLICA DE GNU PARA LA REALIZACIÓN DE COPIAS, DISTRIBUCIÓN Y MODIFICACIÓN</i>	68
8. BIBLIOGRAFÍA.....	75

1. INTRODUCCIÓN

Pensar en la palabra biblioteca, trae a la mente silencio... centenares, miles de libros, el intelecto de tanta gente, gente como yo, como Ud., que en el afán de difundir sus conocimientos, con la intención de enseñar, escribió ese ejemplar de aquel libro tan recordado...

Hoy, la tecnología, los medios de comunicación y la información masiva, nos permiten ver más allá de nuestros horizontes. Es muy divertido, y a la vez interesante, sumergirse en los mares de Internet, navegar en busca de nuestras necesidades y llegar a nuestro objetivo. Lamentablemente, no siempre es posible, los datos están diseminados por muchas partes, hay mucho material... Justamente este aluvión de información hace que haya una sobrecarga de datos, y ya sabemos que los extremos no siempre traen lo mejor...

¿Que tal combinar todo esto? Sería maravilloso lograr fusionar la idea de la vieja biblioteca con la nueva ola de datos. Alcanzar todos los libros, sin importar si alguien los reservó antes, sin interesar si físicamente, no se encuentran en mi biblioteca.

Sabemos del proyecto. Biblioteca 2000 es una realidad, y es mi intención apoyarla.

Una red informática, está formada por un conjunto de computadoras interconectadas de tal forma que permitan compartir sus recursos. Esta es la base fundamental de Biblioteca 2000, compartir.

Es preciso para comprender en detalle este trabajo, que Ud. tenga mínimos conocimientos de computación, y en lo posible, muchas ganas de aprender. Al hablar de red, necesitamos esclarecer un poco ciertos conceptos, disponer de un lenguaje común de comunicación, y en esto, justamente reside el objetivo central de este trabajo.

Muchas preguntas surgen cuando se desarrolla un sistema informático, al trabajar conectados hay que tener en cuenta que no todos los usuarios tienen los mismos derechos, hay personas dedicadas al mantenimiento del sistema y otras que sólo intentan obtener algún provecho de él ocasionalmente. Esto deriva sin dudas en la

palabra seguridad. En contrapartida, se debe tener en cuenta la flexibilidad, la comodidad que brinda el sistema.

Los sistemas UNIX se caracterizan por combinar de forma armónica estos detalles. Son utilizados actualmente en la mayoría de las organizaciones públicas y privadas que necesiten comunicarse de la mejor manera. UNIX, es probablemente el sistema operativo más versátil y popular que se puede encontrar actualmente.

A simple vista podríamos pensar que UNIX es la panacea, bueno, no debería creer esto. Existen varias razones para pensar que UNIX no es lo más adecuado, y una de las más importantes, sino la más importante, son los recursos económicos con que cuenta Biblioteca 2000. Este proyecto, está pensado para la gente, por consiguiente, lo ideal es construir la biblioteca bajo cimientos pensados para la gente.

Podríamos pensar que uno de los cimientos de nuestro sistema, será el tipo de plataforma que utilizaremos en nuestros ordenadores. Lo adecuado sería complementar un sistema tan fuerte como UNIX con una filosofía aún mayor, que es la de dar y recibir. De esto se trata Linux.

Linux es un sistema operativo para PC basados en Intel. Al sistema lo han diseñado y desarrollado cientos de programadores repartidos por todo el mundo. El objetivo ha sido crear un clon de UNIX, sin ningún software comercial con derechos de autor y que pueda utilizar todo el mundo. De hecho, Linux, comenzó como una afición de Linus Torvalds mientras estudiaba en la universidad de Helsinki, en Finlandia. Su objetivo era crear un sustituto del sistema Minix, similar a UNIX.

Linux es un espécimen único en la revolución informática. No se trata de un producto comercial respaldado por una gran empresa, sino más bien de un sistema operativo hijo de la frustración, creado por un equipo de informática de todo el mundo que se ha servido de los recursos de Internet para comunicarse entre sí y crear el sistema operativo denominado Linux.

Como se mencionó anteriormente, la intención de este trabajo es esclarecer conceptos, como así también identificar un lenguaje común para la comunicación. Esta comunicación se dará entre computadores, pues entonces habrá que armar una red, y

si presta atención, aquí encontrará material suficiente para administrar una red Linux de la manera más sencilla.

Qué espera... Manos a la obra!!!

2. BREVE HISTORIA DE LINUX

UNIX es uno de los sistemas operativos más populares del mundo debido a su extenso soporte y distribución. Originalmente fue desarrollado como sistema multitarea con tiempo compartido para miniordenadores y mainframes a mediados de los '70, y desde entonces se ha convertido en uno de los sistemas más utilizados a pesar de su, ocasionalmente, confusa interfaz con el usuario y el problema de su estandarización.

¿Cuál es la verdadera razón de la popularidad de UNIX? Muchos hackers consideran que UNIX es el auténtico y único sistema operativo. El desarrollo de Linux parte de un grupo en expansión de hackers de UNIX que quisieron hacer su sistema con sus propias manos.

Existen numerosas versiones de UNIX para muchos sistemas, desde ordenadores personales hasta supercomputadores como el Cray Y-MP. La mayoría de las versiones de UNIX para ordenadores personales son muy caras. Cuando se escribía este libro, una copia para una máquina 386 del UNIX System V de AT&T costaba unos 1500 dólares estadounidenses.

Linux es una versión de UNIX de libre distribución, inicialmente desarrollada por Linus Torvalds en la Universidad de Helsinki, en Finlandia. Fue desarrollado con la ayuda de muchos programadores y expertos de UNIX a lo largo y ancho del mundo, gracias a la presencia de Internet. Cualquier habitante del planeta puede acceder a Linux y desarrollar nuevos módulos o cambiarlo a su gusto.

El núcleo de Linux no utiliza ninguna línea del código de AT&T o de cualquier otra fuente de propiedad comercial, y buena parte del software para Linux se desarrolla bajo las reglas del proyecto de GNU de la Free Software Foundation, Cambridge, Massachusetts.

Inicialmente, sólo fue un proyecto de Linus Torvalds. Se inspiraba en Minix, un pequeño UNIX desarrollado por Andy Tanenbaum, y las primeras discusiones sobre Linux surgieron en el grupo de News comp.os.minix. Estas discusiones giraban en torno al desarrollo de un pequeño sistema UNIX de carácter académico dirigido a aquellos usuarios de Minix que querían algo más.

El desarrollo inicial de Linux ya aprovechaba las características de conmutación de tareas en modo protegido del 386, y se escribió todo en ensamblador. Linus dice, “Comencé a utilizar el C tras escribir algunos drivers, y ciertamente se aceleró el desarrollo. En este punto sentí que mi idea de hacer un *Minix mejor que Minix* se hacía más seria. Esperaba que algún día pudiese recompilar el gcc bajo Linux... Dos meses de trabajo, hasta que tuve un driver de discos (con numerosos bugs, pero que parecía funcionar en mi PC) y un pequeño sistema de ficheros. Aquí tenía ya la versión 0.01 (al final de Agosto de 1991) no era muy agradable de usar sin el driver de disquetes, y no hacía gran cosa. No pensé que alguien compilaría esa versión.”

No se anunció nada sobre esa versión, puesto que las fuentes del 0.01 jamás fueron ejecutables: contenían solo rudimentos de lo que sería el núcleo, y se asumía que se tenía acceso a un Minix para poderlo compilar y jugar con él.

El 5 de Octubre de 1991, Linus anunció la primera versión “oficial” de Linux, la 0.02. Ya podía ejecutar bash (el shell de GNU) y gcc (el compilador de C de GNU), pero no hacía mucho más. La intención era ser un juguete para hackers. No había nada sobre soporte a usuarios, distribuciones, documentación ni nada parecido. Hoy, la comunidad de Linux aún trata estos asuntos de forma secundaria. Lo primero sigue siendo el desarrollo del kernel.

Linus escribía en comp.os.minix, “¿Suspiráis al recordar aquellos días de Minix-1.1, cuando los hombres eran hombres y escribían sus propios drivers? ¿Os sentís sin ningún proyecto interesante y os gustaría tener un verdadero Sistema operativo que pudieras modificar a placer? ¿Os resulta frustrante el tener solo a Minix? Entonces, este artículo es para vosotros”.

“Como dije hace un mes, estoy trabajando en una versión gratuita de algo parecido a Minix para ordenadores At-386. He alcanzado la etapa en la que puede ser utilizable y voy a poner las fuentes para su distribución. Es solo la versión 0.02, pero he conseguido ejecutar en él: bash, gcc, gnu-make, gnu-sed, compress, etc.”.

Tras la versión 0.03, Linus saltó a la versión 0.10 al tiempo que más gente empezaba a participar en su desarrollo. Tras numerosas revisiones, se alcanzó la versión 0.95, reflejando la esperanza de tener lista muy pronto una versión “oficial”.

(Generalmente, la versión 1.0 de los programas se corresponden con la primera teóricamente completa y sin errores). Esto sucedía en Marzo de 1992. Año y medio después, en Diciembre del 93, el núcleo estaba en la revisión 0.99.pl14, en una aproximación asintótica al 1.0. Actualmente, el núcleo se encuentra en la versión 1.1 parche 52, y se acerca la 1.2.2.

Hoy Linux es ya un clónico de UNIX completo, capaz de ejecutar X Windows, TCP/IP, Emacs, UUCP y software de correo y News. Mucho software de libre distribución ha sido ya portado a Linux, y están empezando a aparecer aplicaciones comerciales. El hardware soportado es mucho mayor que en las primeras versiones del núcleo. Mucha gente ha ejecutado tests de rendimiento en sus sistemas Linux 486 y se han encontrado que son comparables a las estaciones de trabajo de gama media de Sun Microsystems y Digital.

¿Quién iba a imaginar que este “pequeño” clónico de UNIX iba a convertirse en un estándar mundial para los ordenadores personales?

2.1 Características del sistema

Linux implementa la mayor parte de las características que se encuentran en otras implementaciones de UNIX, más algunas otras que no son habituales. En esta sección nos daremos una vuelta por todo ello.

Linux es un sistema operativo completo con multitarea y multiusuario (como cualquier otra versión de UNIX). Esto significa que pueden trabajar varios usuarios simultáneamente en él, y que cada uno de ellos puede tener varios programas en ejecución.

El sistema Linux es compatible con ciertos estándares de UNIX a nivel de código fuente, incluyendo el IEEE POSIX.1, System V y BSD. Fue desarrollado buscando la portabilidad de los fuentes: encontrará que casi todo el software gratuito desarrollado para UNIX se compila en Linux sin problemas. Y todo lo que se hace para Linux (código del núcleo, drivers, librerías y programas de usuario) es de libre distribución.

En Linux también se implementa el control de trabajos POSIX (que se usa en los shells `csch` y `bash`), las pseudoterminales (dispositivos `pty`), y teclados nacionales mediante administradores de teclado cargables dinámicamente. Además, soporta consolas virtuales, lo que permite tener más de una sesión abierta en la consola de texto y conmutar entre ellas fácilmente. A los usuarios del programa “`screen`” les resultará familiar esto.

El núcleo es capaz de emular por su cuenta las instrucciones del coprocesador 387, con lo que en cualquier 386 con coprocesador o sin él se podrán ejecutar aplicaciones que lo requieran.

Linux soporta diversos sistemas de ficheros para guardar los datos. Algunos de ellos, como el `ext2fs`, han sido desarrollados específicamente para Linux. Otros sistemas de ficheros, como el Minix-1 o el de Xenix también están soportados. Y con el de MS-DOS se podrán acceder desde Linux a los disquetes y particiones en discos duros formateados con MS-DOS. Además, también soporta el ISO-9660, que es el estándar en el formato de los CD-ROMs.

Linux implementa todo lo necesario para trabajar en red con TCP/IP. Desde administradores para las tarjetas de red más populares hasta SLIP/PPP, que permiten acceder a una red TCP/IP por el puerto serie. También se implementan PLIP (para comunicarse por el puerto de la impresora) y NFS (para acceso remoto a ficheros). Y también se han portado los clientes de TCP/IP, como FTP, telnet, NNTP y SMTP.

El núcleo de Linux ha sido desarrollado para utilizar las características del modo protegido de los microprocesadores 80386 y 80486. En concreto, hace uso de la gestión de memoria avanzada del modo protegido y otras características avanzadas. Cualquiera que conozca la programación del 386 en el modo protegido sabrá que este modo fue diseñado para su uso en UNIX (o tal vez Multics). Linux hace uso de esta funcionalidad precisamente.

El núcleo soporta ejecutables con paginación por demanda. Esto significa que sólo los segmentos del programa que se necesitan se cargan en memoria desde el disco. Las páginas de los ejecutables son compartidas mediante la técnica `copy-on-`

write, contribuyendo todo ello a reducir la cantidad de memoria requerida para las aplicaciones.

Con el fin de incrementar la memoria disponible, Linux implementa la paginación con el disco: puede tener hasta 256 megabytes de espacio de intercambio o “swap” en el disco duro. Cuando el sistema necesita más memoria, expulsará páginas inactivas al disco, permitiendo la ejecución de programas más grandes o aumentando el número de usuarios que puede atender a la vez. Sin embargo, el espacio de intercambio no puede suplir totalmente a la memoria RAM, ya que el primero es mucho más lento que ésta.

La memoria dedicada a los programas y a la caché de disco está unificada. Por ello, si en cierto momento hay mucha memoria libre, el tamaño de la caché de disco aumentará acelerando así los accesos.

Los ejecutables hacen uso de las librerías de enlace dinámico. Esto significa que los ejecutables comparten el código común de las librerías en un único fichero, como sucede en SunOS. Así, los ejecutables serán más cortos a la hora de guardarlos en el disco, incluyendo aquellos que hagan uso de muchas funciones de librería. También pueden enlazarse estáticamente cuando se deseen ejecutables que no requieran la presencia de las librerías dinámicas en el sistema. El enlace dinámico se hace en tiempo de ejecución, con lo que el programador puede cambiar las librerías sin necesidad de recompilación de los ejecutables.

2.1.1 Multitarea

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

Se le denomina *multitarea prioritaria* porque cada programa tiene garantizada la oportunidad de ejecutarse, y se ejecuta hasta que el sistema operativo da prioridad a otro programa para que se ejecute. Este tipo de multitarea es exactamente lo que hace Linux. MS-DOS y Windows 3.1 no admiten la multitarea prioritaria; admiten una forma de multitarea denominada *multitarea cooperativa*. Con ésta, los programas se ejecutan hasta que permiten voluntariamente que se ejecuten otros programas o no tienen nada

más que hacer por el momento. Para comprender mejor la capacidad multitarea de Linux, examínelo desde otro punto de vista. El microprocesador sólo es capaz de hacer una tarea a la vez, pero las realiza en tiempos tan cortos que se escapan a nuestra comprensión es por eso que en sus “ratos libres” se dedica a ejecutar otras tareas que se le hayan pedido.

Es fácil ver las ventajas de disponer de multitarea prioritaria. Además de reducir el tiempo muerto (tiempo en el que no puede seguir trabajando en una aplicación porque un proceso aún no ha finalizado), la flexibilidad de no tener que cerrar las ventanas de las aplicaciones antes de abrir y trabajar con otras es infinitamente mucho más cómoda.

Linux y otros sistemas operativos multitarea prioritaria consiguen el proceso de prioridad supervisando los procesos que esperan para ejecutarse, así como los que se están ejecutando. El sistema programa entonces cada proceso para que disponga de las mismas oportunidades de acceso al microprocesador. El resultado es que las aplicaciones abiertas parecen estar ejecutándose al mismo tiempo (en realidad, hay una demora de billonésimas de segundo entre el momento en que el procesador ejecuta una serie de instrucciones de una aplicación y el momento programado por Linux para volver a dedicar tiempo a dicho proceso). Es esta capacidad de asignar tiempo a las aplicaciones que se están ejecutando desde una sistema operativo de acceso gratuito, lo que destaca a Linux de otros sistemas operativos y entornos disponibles en la actualidad.

2.1.2 Multiusuario

La idea de que varios usuarios pudieran acceder a las aplicaciones o la capacidad de proceso de un único PC era una utopía hace relativamente pocos años. UNIX y Windows NT ayudaron a convertir ese sueño en realidad. La capacidad de Linux para asignar el tiempo de microprocesador simultáneamente a varias aplicaciones ha derivado en la posibilidad de ofrecer servicio a diversos usuarios a la vez, ejecutando cada uno de ellos una o más aplicaciones. La característica que más resalta de Linux es que un grupo de personas puede trabajar con la misma versión de

la misma aplicación al mismo tiempo, desde el mismo terminal o desde terminales distintos. *No se debe confundir esto con el hecho de que varios usuarios puedan actualizar el mismo archivo simultáneamente, característica que es potencialmente confusa, peligrosa y decididamente indeseable.*

2.2 Diferencias entre Linux y otros sistemas operativos

Es importante entender las diferencias entre Linux y otros sistemas operativos, tales como MS-DOS, OS/2, y otras implementaciones de UNIX para ordenador personal. Primeramente, conviene aclarar que Linux puede convivir felizmente con otros sistemas operativos en la misma máquina: es decir, Ud. puede correr MS-DOS y OS/2 en compañía de Linux sobre el mismo sistema sin problemas.

¿Por qué usar Linux en lugar de un sistema operativo comercial conocido, bien probado, y bien documentado? Podríamos darle miles de razones. Una de las más importantes es que Linux es una excelente elección para trabajar con UNIX a nivel personal. Si Ud. es un desarrollador de software UNIX, Linux le permitirá desarrollar y probar el software UNIX en su PC, incluyendo aplicaciones de bases de datos y X Windows. Si es Ud. estudiante, la oportunidad está en que los sistemas de su universidad correrán UNIX. Con Linux, podrá correr su propio sistema UNIX y adaptarlo a sus necesidades. La instalación y uso de Linux es también una excelente manera de aprender UNIX si no tiene acceso a otras máquinas UNIX.

Pero no perdamos la vista. Linux no es sólo para los usuarios personales de UNIX. Es robusto y suficientemente completo para manejar grandes tareas, así como necesidades de cómputo distribuidas. Muchos negocios, especialmente los pequeños, se están cambiando a Linux en lugar de otros entornos de estación de trabajo basados en UNIX. Las universidades encuentran a Linux perfecto para dar cursos de diseño de sistemas operativos. Grandes vendedores de software comercial se están dando cuenta de las oportunidades que puede brindar un sistema operativo gratuito.

3. INTRODUCCIÓN A LAS REDES

3.1 Historia

La idea de redes es probablemente tan vieja como las telecomunicaciones. Considere a la gente que vivía en la edad de piedra, donde los tambores se pueden haber usado para transmitir mensajes entre individuos. Suponga que el cavernícola A quiere invitar al cavernícola B a un partido de lanzamiento de rocas contra el otro, pero viven demasiado lejos como para que B oiga a A golpear su tambor. ¿Cuáles son las opciones de A?

- ir a la cueva de B.
- hacerse con un tambor más grande.
- pedirle a C, que vive a mitad de camino entre los dos, que le retransmita el mensaje.

La última opción es lo que se llama una red.

Claro que ya ha pasado un tiempo desde las persecuciones y planes primitivos de nuestros antepasados. Hoy en día tenemos ordenadores que hablan entre sí a través de vastas conexiones de cables, fibras ópticas, microondas, etc. A continuación trataremos sobre el significado y modos en que se realiza esto fuera de los cables.

Describiremos dos tipos de redes en esta guía: Las basadas en UUCP, y las basadas en TCP/IP. Estos son conjuntos de protocolos y paquetes de software que suministran medios para transportar datos entre dos ordenadores.

Definiremos una red como un conjunto de nodos que son capaces de comunicarse entre sí, contando con los servicios de un número de nodos dedicados que conmutan datos entre participantes. Los nodos son a menudo ordenadores, pero no es necesario; cualquiera puede pensar también en terminales X o impresoras inteligentes como nodos.

3.2 Redes

Linux dispone de bs dos principales protocolos de red para sistemas UNIX: TCP/IP y UUCP. TCP/IP (para los aficionados a los acrónimos, Transmission Control Protocol/Internet Protocol) es un conjunto de protocolos de red que permite a sistemas de todo el mundo comunicarse en una única red conocida como Internet. Con Linux, TCP/IP y una conexión a la red, puede comunicarse con usuarios y máquinas por toda Internet mediante correo electrónico, noticias (USENET news), transferencias de ficheros con FTP y mucho más. Actualmente hay muchos sistemas Linux conectados a Internet.

La mayoría de las redes TCP/IP usan Ethernet como tipo de red física de transporte. Linux da soporte a muchas tarjetas de red Ethernet e interfaces para ordenadores personales, incluyendo el adaptador Ethernet D-Link de bolsillo para ordenadores portátiles.

Pero dado que no todo el mundo tiene una conexión Ethernet en casa, Linux también proporciona SLIP (Serial Line Internet Protocol), el cual permite conectarse a Internet a través de un módem. Para poder usar SLIP, necesitará tener acceso a un servidor de SLIP, una máquina conectada a la red que permite acceso de entrada por teléfono. Muchas empresas y universidades tienen servidores SLIP disponibles. De hecho, si su sistema Linux dispone de conexión Ethernet y de módem, puede configurarlo como servidor de SLIP para otros usuarios.

NFS (Network File System) permite fácilmente compartir ficheros con otras máquinas de la red. FTP (File Transfer Protocol) permite la transferencia de ficheros entre máquinas.

Si tiene experiencia con aplicaciones TCP/IP en otros sistemas UNIX, Linux le será muy familiar. El sistema proporciona la interface estándar de programación por “sockets”, lo que virtualmente permite que cualquier programa que use TCP/IP pueda ser llevado a Linux. El servidor Linux de X también soporta TCP/IP, permitiendo ver aplicaciones que están corriendo en otros sistemas sobre su pantalla.

UUCP (UNIX-to-UNIX Copy) es un viejo mecanismo usado para transferir ficheros, correo electrónico y noticias entre máquinas UNIX. Clásicamente las

máquinas UUCP conectan entre ellas mediante líneas telefónicas y módem, pero UUCP es capaz de funcionar también sobre una red TCP/IP. Si no tiene acceso a una red TCP/IP o a un servidor SLIP, puede configurar su sistema para enviar y recibir ficheros y correo electrónico usando UUCP.

3.3 La importancia de una administración adecuada

Todos los sistemas UNIX son distintos de un forma u otra y cada sistema individual UNIX es distinto en la forma en que debe administrarse. Linux no es una excepción. Las tareas de administración varían dependiendo, entre otras cosas, del número de usuarios a administrar, los tipos de periféricos conectados al conmutador, las conexiones de red y el nivel de seguridad necesaria.

Un administrador de sistema tiene que proporcionar a los usuarios del sistema un entorno eficiente, seguro y fiable.

La delegación de las responsabilidades de administración varía de un sistema a otro. En sistemas pequeños se asigna a un simple usuario la tarea de administrador. Si se trabaja en un entorno de red, la administración la realiza un administrador de red.

Todos los sistemas Linux tienen un sólo usuario que puede realizar cualquier operación en el computador denominado *superusuario*, con un nombre especial de entrada llamado root (descripto a continuación).

3.3.1 Acerca de raíces, sombreros y la sensación de poder

Como sabe, UNIX distingue diferentes rangos de usuarios. Cada usuario recibe una cuenta que incluye un nombre de usuario y un directorio inicial entre otras cosas. Además de las cuentas dadas a personas reales, existen cuentas especiales, definidas por el sistema, que tienen privilegios especiales. La más importante de éstas es la cuenta raíz, con el nombre de usuario root.

3.3.2 La cuenta root

Los usuarios normales están restringidos normalmente para que no puedan dañar a nadie más en el sistema, sólo a ellos mismos. Los permisos de los ficheros en el sistema están preparados para que los usuarios normales no tengan permitido borrar o modificar ficheros en directorios compartidos por todos los usuarios (como son /bin y /usr/bin). Muchos usuarios también protegen sus propios ficheros con los permisos adecuados para que otros usuarios no puedan acceder o modificar éstos ficheros.

Estas restricciones desaparecen para root. El usuario root puede leer, modificar o borrar cualquier fichero en el sistema, cambiar permisos y pertenencias en cualquier fichero, y ejecutar programas especiales, como pueden ser los que particionan un disco o crean sistemas de ficheros. La idea básica es que la persona o personas que ejecutan y cuidan del sistema entren como root cuando sea necesario para realizar tareas que no pueden ser ejecutadas por un usuario normal. Puesto que root puede hacer todo, es fácil cometer errores que tengan consecuencias catastróficas cuando se trabaja utilizando esta cuenta.

Por ejemplo, como un usuario normal, si inadvertidamente se intentase borrar todos los ficheros en /etc, el sistema no lo permitiría. Sin embargo, como usuario root, el sistema no diría nada. Es muy simple el dañar sistema utilizando root. La mejor forma de evitar accidentes es:

- Pensar dos veces antes de presionar <enter> en un comando que pueda causar daño.
- No acostumbrarse a utilizar root. Cuanto más comfortable se encuentre uno trabajando con el usuario root, más seguro que se confundirán los privilegios con los de un usuario normal.
- Utilizar un marcador distinto para la cuenta root. Se debe cambiar el fichero .bashrc o el login de root para poner el marcador del intérprete a algo distinto del marcador de usuario normal. Por ejemplo, mucha gente utiliza el carácter “\$” como marcador para los usuarios normales, y reserva el carácter “#” como marcador para el usuario root.

- Conectarse como root sólo cuando sea absolutamente necesario. Y desconectarse tan pronto como se haya terminado el trabajo. Cuanto menos se use la cuenta root, menos posibilidades habrá de dañar el sistema.

Por supuesto, existe una variante de hackers de UNIX que utilizan root para prácticamente todo. Pero cada uno de ellos ha cometido, en algún momento, algún error tonto como root y corrompido el sistema. La regla general es que hasta que se familiarice con la falta de restricciones de root, y esté cómodo utilizando el sistema sin dichas restricciones, procure conectarse como root lo menos posible.

Por supuesto, todo el mundo comete errores. El mismo Linus Torvalds borró accidentalmente una vez el árbol de directorios completo del núcleo de su sistema. Horas de trabajo se perdieron para siempre. Afortunadamente, sin embargo, gracias a su conocimiento del código del sistema de ficheros, fue capaz de reiniciar el sistema y reconstruir el árbol de directorios del disco a mano.

Pongámoslo de otra forma, si se imagina la utilización de la cuenta root como un sombrero mágico especial que le da montones de poderes, pudiendo, con un gesto de la mano destruir ciudades enteras, es una buena idea de lo que es ser un poco cauto con lo que se hace con las manos. Puesto que es fácil mover la mano de manera destructiva por accidente, no sería ideal llevar el sombrero mágico cuando no es necesario, a pesar de la maravillosa sensación.

3.4 Comprensión de los sistemas de proceso centralizado

Debido al avance tecnológico durante las décadas de 1950 y 1960, los sistemas operativos comenzaron a permitir que varios usuarios compartieran recursos desde terminales distintos. Dos usuarios podían, en una secuencia de proceso por lotes, ejecutar dos conjuntos de instrucciones al mismo tiempo que compartían un procesador, el almacenamiento y la salida.

Con la llegada de la red telefónica conmutada, los computadores comenzaron a utilizar los recursos telefónicos para extender geográficamente los recursos informáticos. Con este modelo, cada procesador utilizaba recursos de proceso de comunicaciones para conectarse con terminales remotos. Esto hizo necesario que los

terminales y computadores se pudieran comunicar de una forma mejor. El resultado fue el desarrollo del *procesamiento de interfaz de cara al usuario* para las tareas de comunicación y el modelo de procesamiento centralizado.

Hasta que los computadores personales fueron de un precio accesible, potentes, y de uso general, la mayoría de los sistemas UNIX utilizaban el modelo de proceso centralizado. Con este tipo de modelo los computadores mainframe se encargaban de todo el procesamiento. Este modelo cada vez se utiliza menos hoy en día, aunque es adecuado para las ubicaciones informáticas cuyos usuarios estén dispersos geográficamente.

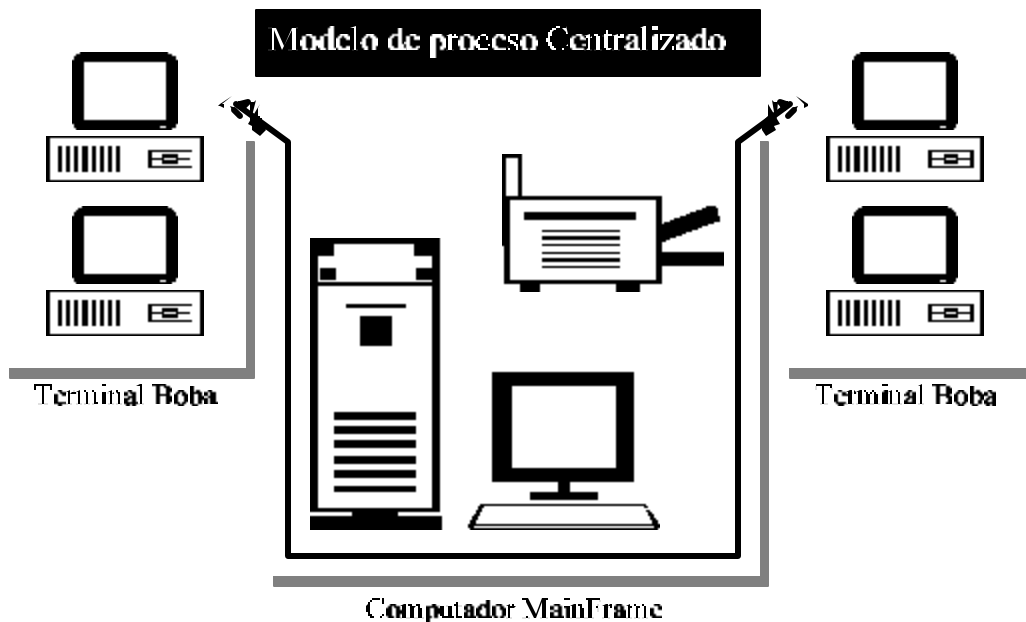


Ilustración 3-1 (Modelo centralizado)

3.4.1 Elementos del modelo de procesamiento centralizado

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

Un servidor puede ser descrito como un computador configurado para compartir sus recursos. Un procesador de interfaz de cara al usuario conecta los canales de

comunicación y el servidor. Se encarga de los detalles de la comunicación de modo que el servidor esté libre para procesar sus datos.

Hoy en día se utilizan dos tipos de terminales populares: *terminales bobas* y *terminales inteligentes*.

Tradicionalmente, UNIX se ha utilizado con los primeros, que sólo cuentan con un teclado y un monitor. No tienen potencia local de proceso. El puerto de comunicación del terminal está conectado al servidor, a través de un módem o directamente. Cuando se escribe en una terminal boba, cada pulsación se transmite al servidor que la procesa.

Los terminales inteligentes, pueden realizar varios procesos mínimos localmente. Las cajas registradoras y otros dispositivos de punto de venta son ejemplos de terminales inteligentes. El dispositivo local, almacena la petición de transacción y la envía completa al servidor en vez de enviar cada pulsación.

Para aumentar el número de puertos disponibles a los que se puede conectar el usuario, hay que instalar un adaptador multipuerto. Por ejemplo, el PC normalmente tiene sólo dos puertos serie: COM1 y COM2. Se necesitan más puertos si se quiere utilizar un PC como servidor para más de dos usuarios.

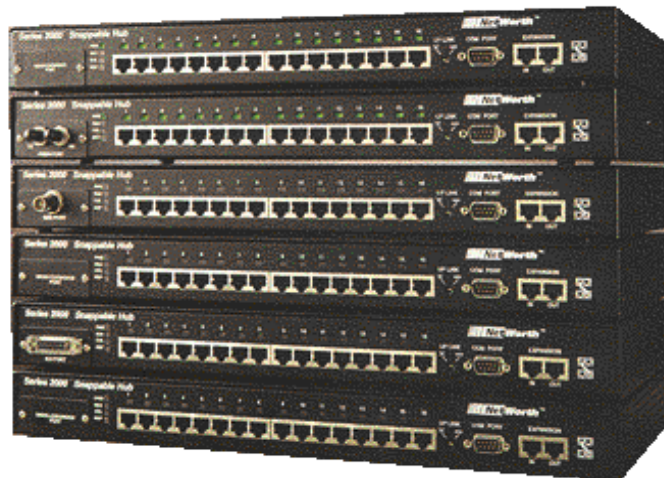


Ilustración 3-2 (Adaptador multipuerto)

3.5 Comprensión de los sistemas de procesamiento distribuido

En el procesamiento distribuido, el terminal se sustituye por una estación de trabajo, que en sí misma es un computador, normalmente ejecutando DOS o UNIX. Los programas pueden estar situados en el servidor, y ejecutarse desde allí, o dentro de la misma estación de trabajo. Se puede imprimir en impresoras conectadas a la estación o en dispositivos distantes conectados al servidor.

3.5.1 Los elementos del modelo de procesamiento distribuido

El procesamiento distribuido utiliza servidores de archivos, estaciones de trabajo, tarjetas de interfaz de red, módulos, repetidores, puentes, routers y pasarelas.

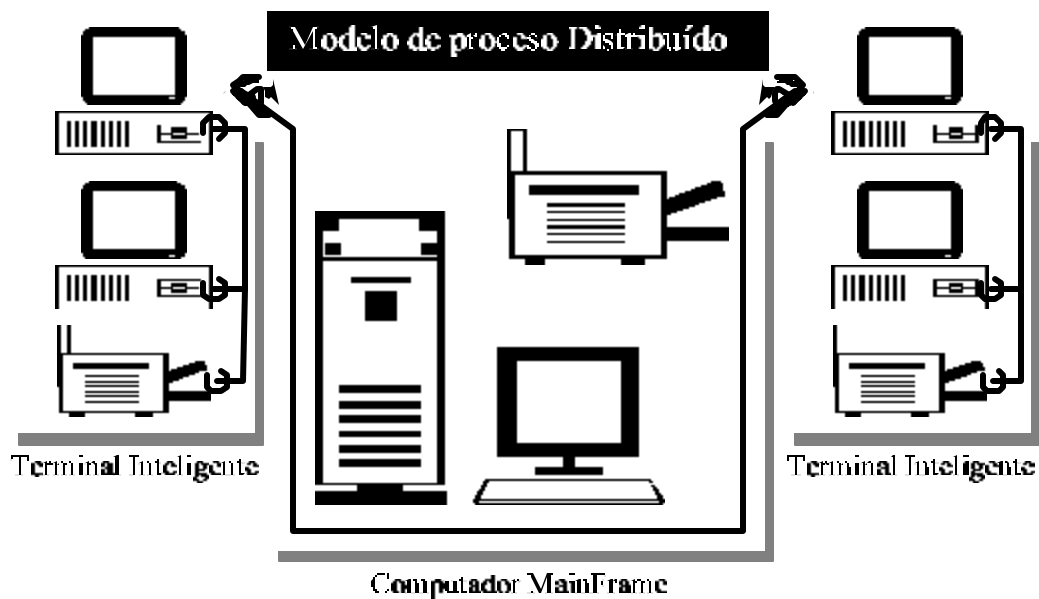


Ilustración 3-3 (Modelo distribuido)

Servidor de archivos: Su propósito es distribuir archivos y segmentos de programas a las estaciones de trabajo, imprimir desde una ubicación central y controlar el flujo de conexión entre las estaciones de trabajo.

Estaciones de trabajo: Un computador personal, además de utilizarse como servidor de archivos, puede también servir como estación de trabajo Linux. Con un procesador 386X y 4Mb de RAM debería ser suficiente para poder trabajar.

Tarjetas de interfaz de red: Una tarjeta de interfaz de red se conecta a una ranura de la placa madre y es el enlace físico entre el computador y el cableado de la red. Estas tarjetas comúnmente están disponibles para cables coaxiales y para par trenzado.

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

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

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

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

Pasarelas: Utilice una pasarela cuando quiera conectar tipos de redes no similares. La pasarela realiza las conversiones de protocolo necesarios de modo que las dos redes se puedan conectar.

3.5.2 Revisión de topologías

La palabra topología, se refiere al modo en que las estaciones de trabajo y los servidores de archivos se conectan en una red. El nombre de varias de las topologías proviene del patrón que hacen los cables después de que se conectan varios terminales, estaciones de trabajo y servidores de archivos.

Las topologías más corrientes son: estrella, bus y anillo. Cuando se utiliza más de una topología en una red, esta se describe como una *red híbrida*.

Topología en estrella: Con esta topología todas las estaciones de trabajo están conectadas a un servidor de archivos o nódulo. Con esta disposición se pueden tener nódulos activos o pasivos.

Topología en bus: En una topología en bus, todas las estaciones de trabajo y servidores de archivos comparten un mismo camino. De hecho, están conectados directamente. La topología en bus, constituye la base de los buses ethernet y del token.

Topología en anillo: Una topología en anillo parece una rueda sin eje. El servidor está conectado en modo bus con las estaciones de trabajo, sin embargo los últimos elementos de la red están conectados juntos hasta formar un bucle cerrado. La topología anillo, utiliza un repetidor, al que IBM llama MAU (Multistation Access Unit). La red token-ring de IBM es un ejemplo de topología en anillo.

Topología híbrida: Una topología híbrida, está compuesta por la integración de redes con distintas topologías, haciendo así, una red global homogénea.

3.6 Modelo Cliente/Servidor

El resultado del desarrollo del procesamiento distribuido es un modelo Cliente/Servidor. Ahora Linux puede utilizarse como el cliente, el servidor o ambos.

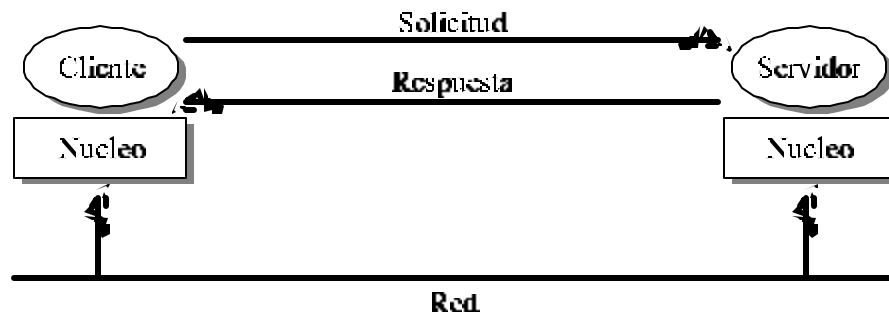


Ilustración 3-4 (Modelo Cliente/Servidor)

Para ilustrar una configuración del modelo Cliente/Servidor, podemos asumir que varias estaciones de trabajo Linux (los clientes) están conectados a un servidor por medio de una topología de bus. El servidor tiene directorios para cada cliente, donde pueden almacenarse los archivos importantes, de los que se hará copia con la copia de seguridad de cada noche. El servidor tiene también directorios donde los clientes pueden compartir archivos. Están conectados al sistema una rápida impresora láser, a la que todos tienen acceso y una unidad de cinta apropiada para hacer copias de

seguridad de los discos grandes. Además, algunos clientes tienen sus propias impresoras, que son más económicas y más lentas que la del servidor.

3.7 Ethernets

El tipo de hardware más usado en las LANs es lo que comúnmente conocemos como Ethernet. Consta de un solo cable con los nodos colgando de él a través de conectores, clavijas o transceptores. Las ethernet simples son poco caras de instalar, lo que unido a un flujo de transferencia neto de 10 Megabits por segundo avala gran parte de su popularidad.

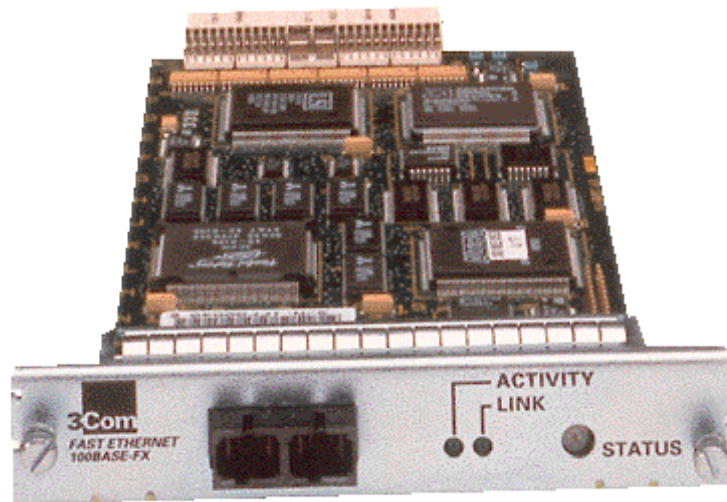


Ilustración 3-5 (Tarjeta Fast Ethernet)

Las Ethernets pueden conectarse mediante tres tipos diferentes de cables: coaxil grueso, coaxil fino y par trenzado. Tanto el fino como el grueso utilizan cable coaxial, difiriendo en el grosor y el modo de conectar este cable a los nodos. El Ethernet fino usa conectores “BNC” con forma de T, que se insertan en el cable y se enganchan en un enchufe en la parte trasera de su ordenador. El Ethernet grueso requiere que taladre un pequeño agujero en el cable ,y conecte un transceptor usando un “conector vampiro”. Entonces se pueden conectar uno o más nodos al transceptor. Los cables Ethernet fino y grueso pueden alcanzar una distancia de 200 y 500 metros, respectivamente, y es por ello que se les llama también 10base-2 y 10base-5.

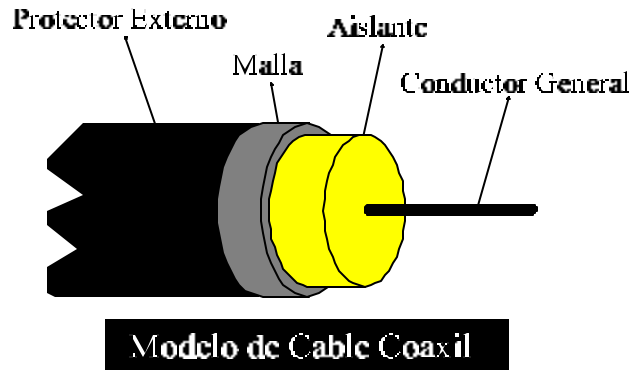


Ilustración 3-6 (Cable coaxil fino)

El par trenzado usa un cable hecho de dos líneas de cobre como las que se encuentran en las instalaciones telefónicas ordinarias, pero generalmente necesitan hardware adicional.

También se le conoce como 10base-T. A pesar de que añadir un nodo a una Ethernet gruesa es una tarea tediosa, eso no tirará abajo la red. Para añadir un nodo en una instalación de cable fino, se debe interrumpir el servicio de red al menos por unos minutos ya que se debe cortar el cable para insertar el conector.

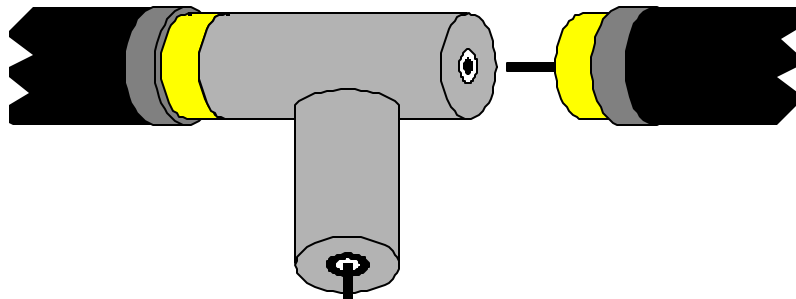


Ilustración 3-7 (Conector tipo BNC)

La mayoría de gente prefiere el Ethernet fino porque es barato: las tarjetas de PC pueden encontrarse por unos \$40, o menos, y el cable esta por unos centavos el metro. Sin embargo, para instalaciones de gran escala, es más apropiado el Ethernet grueso. Por ejemplo, la Ethernet del Departamento de Matemáticas de GMU usa Ethernet gruesa, de modo que no se interrumpen el tráfico cada vez que se añade un nodo a la red.

Uno de los inconvenientes de la tecnología Ethernet es su limitada longitud de cable, que imposibilita cualquier uso fuera de las LANs. Sin embargo, pueden enlazarse varios segmentos de Ethernet entre sí usando repetidores, puentes o routers. Los repetidores simplemente copian las señales entre dos o más segmentos, de forma que todos los segmentos juntos actúan como si fuese una única Ethernet. Los puentes y routers son más sofisticados. Analizan los datos de entrada y los reenvían sólo si el nodo receptor no está en la Ethernet local.

Ethernet funciona como un sistema de bus, donde un nodo puede mandar paquetes (o tramas) de hasta 1500 bytes a otro nodo de la misma Ethernet. Un nodo se direcciona por una dirección de seis bytes grabada en el firmware de su tarjeta Ethernet. Estas direcciones se escriben generalmente como una secuencia de números hexadecimales de dos dígitos separados por dos puntos, como en aa:bb:cc:dd:ee:ff.

Una trama enviada por una estación es visible a todas las estaciones conectadas, pero sólo el nodo destinatario la toma y la procesa. Si dos estaciones intentan emitir al mismo tiempo, se produce una colisión, que se resuelve por parte de las dos estaciones abortando el envío, y reintentándolo al cabo de un tiempo determinado.

3.7.1 Tarjetas Ethernet

Linux soporta un buen número de tarjetas Ethernet y adaptadores para LAN. Esto incluye los siguientes:

- 3com 3c503, 3c503/16
- Novell NE1000, NE2000
- Western Digital WD8003, WD8013
- Hewlett Packard HP27245, HP27247, HP27250
- D-Link DE-600

Los siguientes clónicos se ha informado que funcionan:

- LANNET LEC-45

- Alta Combo
- Artisoft LANtastic AE-2
- Asante Etherpak 2001/2003,
- D-Link Ethernet II
- LTC E-NET/16 P/N 8300-200-002
- Network Solutions HE -203,
- SVEC 4 Dimension Ethernet
- 4-Dimension FD0490 EtherBoard 16

Aquellas tarjetas clónicas compatibles con cualquiera de las tarjetas anteriores también debería funcionar.

3.8 Otros tipos de hardware

En instalaciones mayores, como la Universidad de Groucho Marx, Ethernet no es el único tipo de equipo usado generalmente. En la Universidad de Groucho Marx cada LAN de un departamento esta enlazada a la troncal del campus, que es un cable de fibra óptica funcionando en FDDI (Fiber Distributed Data Interface). FDDI usa una aproximación enteramente diferente para transmitir datos, que básicamente implica el envío de un número de testigos, de modo que una estación sólo pueda enviar una trama si captura un testigo.

La principal ventaja de FDDI es una velocidad de hasta 100 Mbps, y una longitud de cable máxima de hasta 200 km.

Para enlaces de red de larga distancia, se usa frecuentemente un tipo distinto de equipos, que se basa en un estándar llamado X.25. Muchas de las llamadas Redes Publicas de Datos, como Tymnet en Estados Unidos, Datex-P en Alemania, o Iberpac en España, ofrecen este servicio. X.25 requiere un hardware especial, llamado Ensamblador/Desensamblador de Paquetes o PAD. X.25 define un conjunto de protocolos de red de derecho propio, pero sin embargo se usa frecuentemente para conectar redes corriendo TCP/IP y otros protocolos.

Ya que los paquetes IP no se pueden mapear de modo simple en X.25 (y viceversa), estos son encapsulados en paquetes X.25 y enviados a través de la red.

Frecuentemente, los radioaficionados usan sus propios equipos para conectar sus ordenadores en red; esto se llama packet radio o ham radio. El protocolo utilizado por el packet radio es el llamado AX.25, que deriva del X.25. Otras técnicas implican el uso de las lentas pero baratas líneas serie para acceder bajo llamada. Esto requiere otro protocolo para la transmisión de paquetes, como SLIP o PPP, que se describen más adelante.

3.9 Redes con Linux

Siendo el resultado del esfuerzo concentrado de programadores de todo el mundo, Linux no habría sido posible sin la red global. Así que no sorprende que ya en los primeros pasos del desarrollo, varias personas comenzaron a trabajar en dotarlo de capacidades de red. Una implementación de UUCP corría en Linux casi desde el comienzo, y el trabajo sobre redes basadas en TCP/IP comenzó por el otoño de 1992, cuando Ross Biro y otros crearon lo que ahora se conoce como Net-1.

Después de que Ross dejara el desarrollo activo en Mayo de 1993, Fred van Kempen comenzó a trabajar en una nueva implementación, reescribiendo grandes partes del código.

Este esfuerzo continuado se conoce como Net-2. Una primera versión pública, Net-2d, salió en el verano de 1992 (como parte del kernel 0.99.10), y ha sido mantenida y ampliada por varias personas, muy especialmente por Alan Cox, como Net-2 Debugged. Tras una dura corrección y numerosas mejoras en el código, cambió su nombre a Net-3 después de que saliese Linux 1.0. Esta es la versión del código de red que se incluye actualmente en las versiones oficiales del kernel.

Net-3 ofrece drivers de dispositivo para una amplia variedad de tarjetas Ethernet, así como SLIP (para enviar tráfico de red sobre líneas serie), y PLIP (para líneas en paralelo). Con Net-3, Linux tiene una implementación de TCP/IP que se comporta muy bien en entorno de red de área local, mostrándose superior a algunos de los UNIX de

PC comerciales. El desarrollo se mueve actualmente hacia la estabilidad necesaria para su funcionamiento fiable en nodos de Internet.

Además de estas facilidades, hay varios proyectos en marcha que mejorarán la versatilidad de Linux. Un driver para PPP (el protocolo punto a punto, otra forma de enviar tráfico de red sobre líneas serie) está en estado Beta actualmente, y otro driver AX.25 para ham radio está en estado Alfa. Alan Cox también ha implementado un driver para el protocolo IPX de Novell, pero el esfuerzo para un paquete de red completo compatible con el de Novell se ha paralizado por el momento, debido a la negativa de Novell para facilitar la documentación necesaria. Otro proyecto muy prometedor es samba, un servidor de NetBIOS gratis para Unixes, escrito por Andrew Tridge.

3.9.1 Diferentes etapas de desarrollo

Mientras, Fred siguió desarrollando, Net-2e, que dispone de un más revisado diseño de la capa de red. En el momento de escribir esto, Net-2e es aun software Beta. Lo más notable sobre Net-2e es la incorporación de DDI, el Device Driver Interface (Interface de driver de dispositivo). DDI ofrece un acceso y un método de configuración uniforme a todos los dispositivos y protocolos de red.

Otra implementación más de red TCP/IP viene de Matthias Urlichs, quien escribió un driver de RDSI para Linux y FreeBSD. Para esto, integró algo del código de red de BSD en el kernel de Linux.

En un futuro previsible, sin embargo, Net-3 parece estar aquí para quedarse. Alan trabaja actualmente en una implementación del protocolo AX.25 usado por radioaficionados. Sin duda, el código de "módulo" aún por desarrollar para el kernel traerá también nuevos impulsos al código de red. Los Módulos le permiten añadir drivers al kernel en tiempo de ejecución.

Aunque todas estas diferentes implementaciones de red intentan dar el mismo servicio, hay grandes diferencias entre ellas a nivel de kernel y dispositivos. Además, no podrá configurar un sistema con un kernel Net-2e con utilidades de Net-2d o Net-3, y viceversa. Esto solo se aplica a comandos que tienen mucho que ver con el

funcionamiento interno del kernel; aplicaciones y comandos de red comunes como rlogin o telnet se ejecutan en cualquiera de ellos.

3.10 El protocolo IP (Internet Protocol)

Por supuesto, Ud. no querría que su red estuviese limitada a una Ethernet. Idealmente, Ud. desearía poder usar una red sin importarle que hardware dispone y de cuantas subunidades se compone. Por ejemplo, en instalaciones grandes como la Universidad de Groucho Marx, habrá un número de Ethernets separadas, que han de conectarse de alguna manera. En la GMU, el departamento de matemáticas tiene dos Ethernets: una red de máquinas rápidas para profesores y graduados, y otra con máquinas lentas para estudiantes. Ambas están colgadas de la red troncal FDDI del campus.

Esta conexión se maneja con un nodo dedicado, el llamado pasarela, o gateway, que maneja los paquetes entrantes y salientes copiándolos entre las dos Ethernets y el cable de fibra óptica. Por ejemplo, si está en el Departamento de Matemáticas, y quiere acceder a quark en la LAN del Departamento de Físicas desde su Linux, el software de red no puede mandar paquetes a quark directamente, porque no está en la misma Ethernet. Por tanto, tiene que confiar en la pasarela para que actúe como retransmisor. La pasarela (llamémosla sophus) reenvía entonces estos paquetes a su pasarela homóloga niels del Departamento de Físicas, usando la troncal, y por fin niels los entrega a la máquina destino.

Este esquema de dirección de los datos al nodo remoto se llama encaminamiento, y a los paquetes se les denomina a menudo datagramas en este contexto. Para facilitar las cosas, el intercambio de datagramas está gobernado por un único protocolo que es independiente del hardware utilizado.

El principal beneficio del IP es que convierte a redes físicamente distintas en una red aparentemente homogénea. A esto se le llama internet working, y la resultante "metared" se llama internet. Observe aquí la sutil diferencia entre una internet y La Internet. El último es el nombre oficial de una internet global particular. Existen casos

especiales de internets llamadas intranets, las cuáles pueden ser o no parte de La Internet.

Claro que el IP también necesita un esquema de direccionamiento independiente del hardware. Esto se consigue asignando a cada nodo un número único de 32 bits, que define su dirección IP. Una dirección IP se escribe normalmente como 4 números en decimal, uno por cada división de 8 bits, y separados por puntos. Por ejemplo, quark podría tener una dirección IP 0x954C0C04, que se escribiría como 149.76.12.4. A este formato se le llama notación de puntos.

Se dará cuenta de que ahora tenemos tres tipos distintos de direcciones: primero, tenemos el nombre del nodo, como quark, después tenemos las direcciones IP, y por fin están las direcciones hardware, como la dirección Ethernet de 6 bytes. De alguna forma todas ellas deben coincidir, de modo que cuando escriba rlogin quark, se le pueda pasar la dirección IP al software de red; y cuando IP envíe datos a la Ethernet del Departamento de Físicas, de algún modo tiene que encontrar que dirección Ethernet corresponde con la dirección IP. Lo cual es bastante confuso.

No entraremos en esto aquí, sino que lo dejamos para próximos capítulos. De momento, es suficiente con recordar que estos pasos para encontrar las direcciones se llaman resolución de nombres, para mapear nombres de nodo con direcciones IP; y resolución de direcciones, para mapear éstas últimas con direcciones hardware.

3.10.1 Problemas con números IP en Internet

Internet está enfrentando dos graves problemas que coartan su crecimiento. El primero es el rápido agotamiento del ancho de banda de los backbones. Afortunadamente, este problema está en vías de ser resuelto, agregando vínculos de mayor velocidad en la “columna vertebral” de la red de redes.

El segundo problema es más grave, ya que no existe una solución que no implique grandes cambios en los protocolos que utiliza Internet. Como sabemos, todo equipo conectado a una red TCP/IP como Internet debe poseer una dirección de IP única para toda la red. Esta dirección es de 32 bits de largo y generalmente se la representa con cuatro bytes separados por puntos (ejemplo: 128.12.14.5). Si bien con

este esquema son disponibles 4.000 millones de direcciones IP, las direcciones se agruparon en redes de distinto tipo. Una red de clase A posee 16 millones de hosts; una de clase B, 65.000; y una de clase C, 254.

Esta división, si bien ayudaba al enrutamiento de los paquetes TCP/IP, no utilizaba muy eficientemente todas las direcciones posibles. Si bien este esquema resistió satisfactoriamente los primeros 25 años de Internet, desde hace dos años están comenzando a escasear los números de red. Las redes clase B están casi agotadas, mientras que las de clase C van en camino de correr la misma suerte en muy poco tiempo.

La solución a este problema es la adopción de una nueva versión para el protocolo IP, denominada IPv6. El actual protocolo IP, versión 4 (IPv4), soporta direcciones de 32 bits de longitud. IPv6 lleva esta longitud a 128 bits, dando un espacio de direccionamiento de tal magnitud, que se espera que dure mucho más que el anterior. Además, se definieron mecanismos de separación de redes mucho más eficientes. IPv6 no solo aumenta el tamaño de la dirección IP, sino que incorpora soporte de seguridad, transmisores de información sensibles al retardo (como ser audio y video), reserva de ancho de banda, etc.

Si bien un nodo corriendo IPv6 puede conectarse transparentemente con un IPv4, y ya hay varias implementaciones de software para IPv6, tanto para computadoras como para ruteadores; existe una gran preocupación acerca de cómo se va a manejar la transición y si los usuarios de Internet aceptarán el cambio sin inconvenientes. Lamentablemente, el tiempo se acaba y la adopción de IPv6 debe ser efectuada lo más pronto posible.

4. IP EN LÍNEAS SERIE, SLIP Y PPP

SLIP (Serial Line IP, Protocolo Internet en Línea Serie), y PPP (Point-to-Point Protocol, Protocolo Punto-a-Punto) son protocolos muy utilizados para enviar paquetes IP a través de enlaces serie. Varias instituciones ofrecen acceso telefónico SLIP y PPP a maquinas conectadas a Internet, esto proporciona conectividad IP a personas privadas (algo que de otra forma seria difícilmente de conseguir debido al elevado coste).

Para trabajar con SLIP o PPP, no son necesarias modificaciones en el hardware; puede utilizarse cualquier puerto serie.

4.1 Configurando el software serie

Hay personas fuera del mundo de las redes que solo poseen un PC y no tienen dinero para gastarlo en un enlace Internet T1. Para conseguir su dosis diaria de noticias y mensajes, dependen de enlaces SLIP, redes UUCP y BBS, que usan las redes telefónicas publicas.

Este capitulo pretende ayudar a todas aquellas personas que dependen del módem para mantener sus enlaces. Sin embargo, hay muchos detalles que este capitulo no puede abordar, por ejemplo, como configurar el módem para marcar ya que no todos los módems tienen una misma configuración. Por otro lado, nuestra tarea es comprender la administración de redes, sería un trabajo muy extenso abordar temas tan específicos. Para más información, puede consultar "serial HOWTO" de Greg Hankins que es enviado a comp.os.linux.announce regularmente.

4.1.1 Software de comunicaciones vía módem

Existen varios paquetes de comunicaciones disponibles para Linux. Muchos de ellos son emuladores de terminales, que permiten a un usuario conectarse a otro ordenador como si estuviera frente a uno de sus terminales. El emulador de terminales tradicional en UNIX es kermi. Este es, sin embargo, algo duro. Hay programas disponibles más cómodos que soportan agenda telefónica y guiones

para llamar y entrar en ordenadores remotos. Uno de estos es el minicom, muy parecido a los primitivos programas emuladores de terminal a los que tan acostumbrados están los usuarios del DOS. Hay también paquetes de comunicaciones bajo X-11 como por ejemplo seyon.



Ilustración 4-1 (Módem tipo tarjeta)

También, existe un buen numero de programas para instalar BBS bajo Linux, disponibles para aquellos que quieran ofrecer este sistema.



Ilustración 4-2 (Módem tipo externo)

A parte de los programas de terminal, hay también software que usa la línea serie de forma no interactiva para el transporte de datos hasta su ordenador. La ventaja de esta técnica es que se usa menos tiempo en transferir varios kilobytes automáticamente que el necesario para leer los mensajes en algunos buzones y visualizar noticias interesantes en un BBS. Por otra parte, esto requiere más

espacio de disco debido a la transferencia de más información inútil de la que normalmente se obtendría.

El compendio de esta clase de software de comunicaciones es UUCP. Este es un conjunto de programas que copia ficheros desde una maquina a otra, ejecuta programas en un ordenador remoto, etc. Se utiliza frecuentemente para transferir mensajes y noticias (news) entre redes privadas. Otro software de comunicaciones no interactivo es utilizado, por ejemplo, en Fidonet. También están disponibles puertos para aplicaciones de Fidonet como ifmail.

4.2 Introducción a dispositivos serie

Los dispositivos proporcionados por núcleo UNIX para el acceso a dispositivos serie son llamados normalmente ttys. Esta es una abreviatura de Teletype, los cuales solían ser uno de los mayores productores de terminales en los primeros días de UNIX. El termino se usa actualmente para cualquier terminal de texto. A través de este capítulo, lo usaremos exclusivamente para referirnos a los dispositivos.

Linux distingue tres clases de ttys: consolas (virtuales), pseudo terminales (similares a las tuberías de doble vía, usadas por aplicaciones tales como X11), y dispositivos serie. Estos últimos son considerados también como ttys, porque permiten sesiones interactivas sobre conexiones serie, ya sea esta un terminal conectado por cable o un ordenador remoto a través de la línea telefónica.

Los Ttys tiene un numero de parámetros configurables mediante la llamada al sistema ioctl. Muchas de ellas se aplican solo a dispositivos serie, puesto que necesitan una distribución mayor y más flexible para manejar varios tipos de conexiones.

Entre los más destacados parámetros para la línea se encuentra la velocidad y la paridad. Pero hay también elementos para la conversión entre caracteres en mayúscula y minúscula, de retorno de carro en el avance de línea, etc. El manejador tty puede también soportar varias líneas dedicadas las cuales hacen que el manejador de dispositivo tenga un comportamiento completamente distinto. Por

ejemplo, el driver SLIP para Linux está desarrollado por medio de una línea especial dedicada.

Hay una pequeña ambigüedad sobre la forma de medir la velocidad de línea. El termino correcto es Bit rate, el cual esta relacionado con la velocidad de transferencia de la línea medida en bit por segundo (bps para abreviar). Algunas veces se oye a la gente referirlo como velocidad en Baudios, el cual no es muy correcto. Esos dos términos no son intercambiables. La velocidad en Baudios se refiere a una característica física de algunos dispositivos serie, es decir, la velocidad de reloj a la que se transmiten los impulsos. El "bit rate", indica el estado actual de una conexión serie existente entre dos puntos, a saber, el número medio de bits transferidos por segundo. Es importante saber que estos dos valores suelen ser diferentes, ya que la mayoría de los dispositivos codifican más de un bit por cada impulso eléctrico.

4.3 Accediendo a los dispositivos serie

Como todos los dispositivos en un sistema UNIX, se accede a los puertos serie a través de ficheros especiales, localizados en el directorio /dev. Hay dos tipos de ficheros de dispositivos relacionados con los manejadores serie, y cada puerto tiene su archivo. Dependiendo del fichero por el que se acceda, el dispositivo se comportara de forma diferente.

El primer tipo se utiliza para las llamadas de entrada, tienen el 4 como número mayor, y los ficheros son nombrados ttyS0, ttyS1, etc. El segundo tipo se utiliza para llamadas de salida a través de un puerto, los ficheros son llamados cua0, etc. y tienen como número mayor el 5.

Los números menores son idénticos en ambos tipos. Si tiene su módem en uno cualquiera de los puertos COM1 a COM4, su número menor será el numero de puerto COM + 63.

Asumamos que su módem esta en el COM2, en ese caso su numero menor será 65, y su numero mayor será 5 para llamadas. Para esto, debería existir un

dispositivo que tuviera esos números. A continuación vemos una lista de ttys serie del directorio /dev.

```
$ls -l/dev/cua*  
  
crw-rw-rw-1 root root 5, 64 Nov 30 19:31 /dev/cua0  
  
crw-rw-rw-1 root root 5, 65 Nov 30 22:08 /dev/cua1  
  
crw-rw-rw-1 root root 5, 66 Oct 28 11:56 /dev/cua2  
  
crw-rw-rw-1 root root 5, 67 Mar 19 1992 /dev/cua3
```

Si no hubiese un dispositivo tal, tendría que crear uno, haciéndose superusuario y tecleando:

```
#mknod -m 666 /dev/cua1 c 5 65  
  
#chown root.root /dev/cua1
```

4.4 Hardware serie

Linux normalmente soporta una amplia variedad de placas serie que utilicen el estándar RS-232. RS-232 es, en la actualidad, el estándar más común para comunicaciones serie en el mundo del PC. Este usa un número de circuitos tanto para transmitir simples bits como para sincronización. Líneas adicionales pueden ser usadas para señalar la presencia de un carrier y handshake.

El hardware handshake es opcional, pero muy útil. Permite a cada una de las dos estaciones señalar cuando esta lista para recibir más datos, o si la otra estación debe parar hasta que el receptor procese los datos de entrada. Las líneas usadas para esto son llamadas Clear to Send, despejado para enviar, (CTS) y Ready to Send, listo para enviar (RTS), respectivamente.

En ordenadores PC, el interfaz RS-232 es generalmente manejado por un chip UART descendiente del chip 16450 de la National Semiconductor, o de una nueva versión de este, el NSC 16550A2. Algunas marcas (más notablemente los módems internos equipados con un chip Rockwell) también usan chips completamente diferentes que han sido programados para comportarse como un 16550.

La principal diferencia entre los 16450 y los 16550 estriba en que el primero tiene un buffer de 1 byte mientras que el segundo es de 16 bytes. Esto hace al 16450 válido para velocidades máximas de 9600 baudios, mientras que para velocidades superiores se requiere un chip compatible con el 16550. Además de esos chips, Linux también soporta el chip 8250, que era el original UART de los PC.

En la configuración por defecto, el núcleo comprueba los cuatro puertos serie estándar, es decir, del COM1 hasta el COM4, a los que se le asignaran los números menores desde el 64 hasta el 67, como se ha descrito anteriormente.

Si quiere configurar su puerto serie adecuadamente, tendría que incluir la orden setserial en el archivo rc.serial, el cual es llamado en tiempo de arranque del sistema desde el fichero de inicialización /etc/rc. Este usa setserial para configurar los dispositivos serie del kernel. Un típico guión rc.serial tendrá el siguiente aspecto:

```
#/etc/rc.serial - guión de configuración de la línea serie

#

# Detección de interrupciones libres

/sbin/setserial -W /dev/cua*

# Configurar dispositivos serie

/sbin/setserial /dev/cua0 auto_irq skip_test autoconfig

/sbin/setserial /dev/cua1 auto_irq skip_test autoconfig

/sbin/setserial /dev/cua2 auto_irq skip_test autoconfig

/sbin/setserial /dev/cua3 auto_irq skip_test autoconfig

# Muestra la configuración de dispositivos serie

/sbin/setserial -bg /dev/cua*
```

Referirse, por favor, a la documentación que viene con setserial para comprender los parámetros. Si su tarjeta serie no es detectada, o la orden setserial -bg muestra una configuración incorrecta, tendrá que forzar la configuración suministrando explícitamente los valores correctos. Está comprobado que los módems internos equipados con los chip de Rockwell experimentan este tipo de problemas. Así, por ejemplo, si se obtiene que el chip de una UART es el NSC

16450, siendo de hecho una compatible NSC 16550, se tendrá que cambiar la configuración del puerto implicado de la forma siguiente:

```
/sbin/setserial /dev/cua1 auto_irq skip_test autoconfig UART 16550
```

Existen opciones similares para forzar los puertos COM, direcciones base, y configuración de IRQ.

Si su módem soporta hardware handshake, asegúrese de activarlo. Sorprendentemente, la mayoría de los programas de comunicaciones no intentan activarlo por defecto, por lo cual hay que realizarlo manualmente. La mejor forma es incluirlo en el guión rc.serial usando la orden stty:

```
$ stty crtscts < /dev/cua1
```

Para comprobar si el hardware handshake esta funcionando usar:

```
$ stty -a < /dev/cua1
```

Esto le da el estatus de todos los parámetros de ese dispositivo. Un parámetro precedido con un signo menos como en -crtscts significa que ha sido desactivado.

5. EL PROTOCOLO DE CONTROL DE TRANSMISIÓN, TCP

Por otro lado, la historia no se acaba con el envío de datagramas de un nodo a otro. Si accede a hostquark, querrá disponer de una conexión fiable entre su proceso rlogin en *erdos* y el proceso de shell en quark. Para ello, la información enviada en uno y otro sentido debe dividirse en paquetes en el origen, y reensamblada en un flujo de caracteres por el receptor. Parece trivial, pero implica un número de tareas complejas.

SLIP está descrito en la norma RFC 1055. La compresión de cabeceras CSLIP, basada en él, se describe en la RFC 1144.

Una cosa importante a saber sobre IP es que por sí sólo, no es fiable. Suponga que diez personas de su Ethernet comienzan a transferirse la última versión de XFree86 del servidor de FTP de GMU. La cantidad de tráfico generada por esto podría ser excesiva para que la maneje la pasarela, porque es demasiado lento, y anda escaso de memoria. Ahora, si coincide que Ud. envía un paquete a quark, sophus podría tener agotado el espacio del buffer durante un momento y por tanto no es capaz de reenviarlo. IP resuelve este problema simplemente descartándolo. El paquete se pierde irrevocablemente. Lo cual traslada la responsabilidad de comprobar la integridad y exactitud de los datos a los nodos extremos, y su retransmisión en caso de error. De esto se encarga otro protocolo, TCP, o Transmission Control Protocol (Protocolo de Control de la Transmisión), que construye un servicio fiable por encima de IP.

La propiedad esencial de TCP es que usa IP para darle la impresión de una conexión simple entre dos procesos en su equipo y la máquina remota, de modo que no tiene que preocuparse de cómo y sobre qué ruta viajan realmente sus datos. Una conexión TCP funciona básicamente como una tubería de doble sentido en la que ambos procesos pueden escribir y leer. Imagínela como una conversación telefónica.

TCP identifica los extremos de tal conexión por las direcciones IP de los dos nodos implicados, y el número de los llamados puertos de cada nodo. Los puertos se pueden ver como puntos de enganche para conexiones de red. Si vamos a explotar el ejemplo del teléfono un poco más, uno puede comparar las direcciones IP con los

prefijos de área (los números representarían ciudades), y los números de puerto con los códigos locales (números que representan teléfonos de personas concretas).

En el ejemplo de rlogin, la aplicación cliente (rlogin) abre un puerto en *erdos*, y se conecta al puerto 513 de hostquark, en el que se sabe que está escuchando el servidor rlogind.

Esto establece una conexión TCP. Usando esta conexión, rlogind realiza el procedimiento de autorización, y entonces muestra el shell. La entrada y salida estándar del shell se redirigen a la conexión TCP, de modo que cualquier cosa que escriba a rlogin en su máquina será pasado a través del canal TCP y entregado al shell como entrada estándar.

TCP no es el único protocolo de usuario en redes TCP/IP. Aunque adecuado para aplicaciones como rlogin, la sobrecarga que supone es prohibitiva para aplicaciones como NFS. En contraposición, éste usa un protocolo derivado de TCP llamado UDP, o User Datagram Protocol (Protocolo de datagrama de Usuario). De igual modo que TCP, UDP también permite que una aplicación contacte con un servicio en un puerto concreto de la máquina remota, pero no establece una conexión para ello. En cambio, puede usarlo para enviar paquetes sueltos al servicio destino. De allí su nombre.

Suponga que ha montado la jerarquía del directorio TEX del servidor de NFS central del departamento *galois*, y desea ver un documento que describe como usar LATEX. Arranca su editor, y lee el fichero completo. Sin embargo, le llevaría demasiado tiempo establecer una conexión TCP con *galois*, enviar el fichero, y liberarla de nuevo. En cambio, si se hace una petición a *galois*, envía el fichero en un par de paquetes UDP, que es mucho más rápido. Sin embargo, UDP no se hizo para controlar la pérdida o corrupción de paquetes. Es responsabilidad de la aplicación (en este caso NFS) tener en cuenta esto.

Linux soporta una implementación completa de los protocolos de red TCP/IP (Transport Control Protocol/Internet Protocol). TCP/IP ha resultado ser hasta ahora el mejor mecanismo de comunicación entre ordenadores de todo el mundo. Con Linux y

una tarjeta Ethernet podrá introducir su máquina en una red local o (si se tienen las conexiones apropiadas) a la internet, la red TCP/IP de ámbito mundial.

Poner en marcha una pequeña red local de máquinas UNIX es fácil. Sólo requiere una tarjeta Ethernet en cada máquina y los cables adecuados así como hardware accesorio (terminadores, etc.). Y si su universidad o empresa tiene acceso a la internet, podrá insertar su máquina Linux en esta red.

La implementación actual de TCP/IP y los protocolos relacionados para Linux se llama "NET-2". No tiene que ver con la versión NET-2 para BSD. En realidad, se refiere a que es la segunda implementación que se hace para Linux.

NET-2 de Linux soporta también SLIP (Serial Line Internet Protocol). SLIP le permitirá acceder a la Internet con un módem. Si su universidad o empresa proporciona accesos por SLIP, podrá llamar desde su casa al servidor SLIP y conectarse así a la Red. Recíprocamente, si posee en Linux una tarjeta de Red y un módem podrá configurar un servidor SLIP en él.

Para obtener más información de configuración de TCP/IP en Linux, puede leer el documento NET-2 HOWTO, disponible mediante FTP en <http://www.infor.es/LINUX>. Se trata de una guía completa de configuración, que incluye conexiones mediante Ethernet y SLIP. Otro documento relacionado es el Ethernet HOWTO, que se centra en cómo configurar diversos modelos de tarjetas Ethernet. Encontrará más ayuda en el libro TCP/IP Network Administration, de Craig Hunt.

5.1 Hardware requerido

Puede utilizar el TCP/IP para Linux sin hardware de red. Así podrá usar el dispositivo "loopback" para conectarse con usted mismo. Aunque parezca poco serio, hay algunos programas que necesitan conexiones de red "loopback" para funcionar.

Si desea utilizar Linux en una red de tipo Ethernet, necesitará una de las placas citadas anteriormente.

5.2 Configuración de TCP/IP

En esta sección intentaremos explicar cómo configurar una conexión TCP/IP con Ethernet. Nótese que este método funcionará en muchos sistemas, pero no siempre. Nuestra explicación debería ser suficiente para aclararle el camino en la configuración de red en su máquina.

En primer lugar, vamos a asumir que su sistema Linux ha sido instalado con el software TCP/IP. Esto incluye clientes como telnet y ftp, comandos de administración como ifconfig y route (que suelen estar en /etc) y ficheros de configuración de red, como /etc/hosts. También vamos a suponer que el núcleo está compilado con el soporte TCP/IP.

Una vez hecho esto, se deben modificar los ficheros de configuración que usa NET-2. Esta parte suele ser bastante simple, pero existen **desacuerdos** entre las diferentes distribuciones de Linux. Los ficheros pueden estar en /etc o en /usr/etc o incluso /usr/etc/inet. En el peor caso puede usar el comando find para localizar los ficheros. A veces los ficheros están también repartidos por varios directorios y no en uno sólo.

5.2.1 La configuración de red

Antes de configurar su sistema con TCP/IP necesita conocer cierta información sobre la red. En muchos casos, el administrador local se la proporcionará.

Dirección IP: Es la dirección única de cada máquina, formada por números separados por puntos. Por ejemplo, 128.253.153.54. El administrador de red le dará este número. Si está configurando el modo "loopback" únicamente (esto es, no tiene conexión a la red mediante SLIP o Ethernet) su dirección IP será la 127.0.0.1.

Máscara de red: Es un número similar a la dirección IP, que determina en qué parte de la dirección IP se encuentra el número de subred, y qué parte especifica el host en la subred. La máscara de red es un patrón de bits, que al ser superpuesto a una dirección de la red, le dirá en qué subred se encuentra esa dirección. Esto es muy importante para el rutado, y si usted nota que puede comunicarse con gente de redes

externas pero no con gente de su misma red, es posible que tenga mal configurada la máscara de red. Los administradores de la subred habrán seleccionado las máscaras en tiempo de diseño de la red, y serán quienes deban darle esa información.

Muchas subredes son de “clase C” y usan la máscara 255.255.255.0. Otras subredes de “clase B” usan la 255.255.0.0. El código de NET-2 seleccionará automáticamente una máscara que asume que no hay subred. Todo esto debe aplicarse también a la configuración “loopback”. Dado que la dirección “loopback” es siempre la 127.0.0.1, la máscara será la 255.0.0.0. Puede especificarla de forma explícita o dejar que el sistema la ponga por defecto.

Dirección de red: Es el resultado de la operación lógica AND entre su dirección IP y la máscara. Por ejemplo, si su dirección IP es la 128.253.154.32 y la máscara es 255.255.255.0, su dirección de red será la 128.253.154.0. Con una máscara 255.255.0.0, la dirección sería 128.253.0.0. Si utiliza solo la configuración en “loopback”, la dirección de red no existe.

Dirección de “broadcast”: Se utiliza para lanzar paquetes que deben recibir todas las máquinas de la subred. Así pues, si el número de host de la subred se obtiene mediante el último octeto de la dirección IP (o sea, la máscara es la 255.255.255.0), su dirección de “broadcast” será su dirección de red operado en OR con 0.0.0.255. Por ejemplo, si su número IP es el 128.253.154.32, y la máscara es la 255.255.255.0, la dirección de “broadcast” sería la 128.253.154.255. Observe que por motivos históricos, algunas subredes están configuradas para usar la dirección de red como dirección de “broadcast”. Si tiene dudas, consulte con el administrador de la red.

En muchos casos, bastará con copiar la configuración que tengan otras máquinas de la subred y cambiar únicamente el valor IP, por supuesto. La dirección “broadcast” tampoco tiene utilidad en una configuración en “loopback”.

Dirección de pasarela: En ocasiones puede tener varias pasarelas. Una pasarela o gateway es simplemente una máquina que se encuentra a la vez en dos subredes (tiene una dirección IP por cada una), y reparte los paquetes entre ellas. En muchas subredes existe una sola pasarela para comunicarse con las redes externas, pero en otras hay varias, una para cada subred adicional. Si su red está aislada de

otras, o su máquina se encuentra en configuración “loopback”, no necesitará dirección de pasarela.

Se trata de la dirección de la máquina que va a ser su pasarela a otras máquinas que no estén en su misma subred. Muchas veces es una dirección IP como la suya, sólo que terminada en “.1”. Por ejemplo, si su dirección IP es la 128.253.154.32, la de la pasarela podría ser la 128.253.154.1. El administrador se la dirá en cualquier caso.

Dirección del servidor de nombres: Suele existir un servidor que traduce nombres de máquinas a direcciones IP. Puede usted mismo ejecutar en su máquina un servidor de nombres, el programa named, en cuyo caso su dirección será la 127.0.0.1. A menos que realmente lo necesite, le recomendamos que procure siempre usar otra máquina distinta. La configuración de named es otro tema; y lo primordial aquí es que se comunique con la red. Puede tratar estos asuntos más tarde. En una configuración “loopback” no es necesario este dato.

Nota para usuarios de SLIP: La información anterior puede necesitarla o no. Cuando use SLIP su dirección IP será determinada de dos formas: bien “estática”, lo que significa que será siempre la misma, o bien “dinámica”, lo que indica que le será asignada una de las disponibles cada vez que conecte con el servidor SLIP.

NET-2 implementa rutado completo, múltiples rutas, subredes... Lo anterior describe las configuraciones más básicas. Pero la suya puede ser diferente: cuando tenga alguna duda, consulte al administrador de la red, y eche un vistazo a las páginas del manual para route e ifconfig. La configuración completa de redes TCP/IP supera ampliamente las intenciones de este trabajo. Para más información puede consultar “Redes globales de información con Internet y TCP/IP” de Douglas E. Comer.

5.3 Introducción al DNS

DNS organiza los nombres de los nodos en una jerarquía de dominios. Un dominio es una colección de nodos relacionados de alguna manera. Por ejemplo, las universidades norteamericanas se agrupan en el dominio edu, y cada universidad mantiene un subdominio dentro de edu. Nuestro ejemplo, la Universidad de Groucho Marx, mantendría el dominio gm.u.edu y las máquinas del departamento de

Matemáticas se encontrarían dentro del dominio maths.gmu.edu. De modo que el nombre completo de la máquina *erdos* será erdos.maths.gmu.edu. El nombre completo se conoce como nombre totalmente cualificado o FQDN, e identifica a ese nodo en todo el mundo.

Dependiendo de su localización en la jerarquía, un dominio puede ser de primer, segundo o tercer nivel. Otros niveles pueden existir pero no son frecuentes. Por ejemplo, algunos dominios de primer nivel muy usuales son los siguientes:

edu: Aquí se incluyen casi todas las universidades o centros de investigación.

com: Compañías u organizaciones con fines comerciales.

org: Organizaciones no comerciales. Las redes UUCP privadas se encuentran aquí.

net: Pasarelas y otros nodos administrativos de la red.

mil: Nodos militares.

gov: Nodos del gobierno.

Fuera de los Estados Unidos, existe un dominio de primer nivel para cada país, de dos letras según se define en la norma ISO-3166. Finlandia, por ejemplo, usa el dominio **fi**, el dominio **de**, corresponde a Alemania y el dominio **ar** corresponde a Argentina. Cada país organiza por debajo del primer nivel, los dominios de segundo nivel, de manera parecida a los norteamericanos (por ejemplo, con dominios com.ar o edu.ar).

Por supuesto, un nodo dentro del dominio de un país puede no estar físicamente en él. El dominio únicamente identifica al nodo como registrado en el NIC de ese país. Así, un comerciante sueco puede tener una delegación en Australia, y tener sus nodos australianos registrados dentro del dominio de primer nivel sueco, **se**.

Esta organización por dominios soluciona el problema de la unicidad de nombres. Además, los nombres totalmente cualificados no son difíciles de recordar.

Pero DNS tiene otras ventajas: permite delegar la autoridad sobre un determinado subdominio a sus administradores. Por ejemplo, los subdominios maths y

physics de la UGM son creados y mantenidos por el Centro de Cálculo de dicha universidad. Y si el mantenimiento del subdominio `maths.gmu.edu` fuese complicado (por número elevado de nodos, existencia de subdominios internos, etc), el Centro de cálculo de la UGM puede delegar la autoridad sobre ese subdominio al departamento de Matemáticas. La delegación de un subdominio implica el control total del mismo por parte de la organización en la que se delegó, con total libertad para crear nuevos subdominios internos, asociar nombres a nodos, etc.

Para este fin, el espacio de nombres se divide en zonas, cada una asociada a un dominio. Notar que existe una diferencia entre zona y dominio: el dominio `groucho.edu` incluye todos los nodos de la UGM, mientras que la zona `groucho.edu` incluye sólo los nodos que mantiene directamente el Centro de Cálculo, ya que los nodos del subdominio `physics.groucho.edu` pertenecen a la zona controlada por el Departamento de Físicas.

5.3.1 Búsquedas de nombres con DNS

Trataremos aquí el problema de cómo localizar el nombre de un determinado nodo. DNS es una gigantesca base de datos distribuida. Se implementa a través de los llamados servidores de nombres. Cada uno de éstos mantiene la información de uno o varios dominios. Para cada zona hay al menos dos (o más) servidores de nombres que mantienen información autorizada sobre los nodos de esa zona. Para obtener la dirección IP del nodo *erdos*, lo que hay que hacer es contactar con el servidor de nombres de la zona para `groucho.edu` y este nos devolverá los datos pedidos.

Esto parece fácil de decir pero difícil de implementar pues nos preguntaremos cómo localizar al servidor de nombres de la UGM. Si su ordenador no implementa un adivino, le ayudara el DNS. Cuando su aplicación desea encontrar información acerca de *erdos*, contactará en primer lugar con un servidor de nombres local, quien realizará una búsqueda por otros servidores.

Empieza por preguntar a un servidor de nombres raíz por *erdos.maths.groucho.edu*. Al comprobar este último que él no mantiene ese dominio,

contactará con los servidores del dominio edu y les preguntará las direcciones de los servidores de nombres, que retornará al servidor local. Ahora nuestro servidor preguntará a estos últimos y éstos a su vez irán haciendo llegar a nuestro servidor hasta los que mantienen la zona groucho.edu. Finalmente, se preguntará a uno de estos últimos por el nodo erdos y se enviará la respuesta al usuario.

Aparentemente esto provoca mucho tráfico, aunque en todo caso siempre será menor que preguntar siempre a los mismos servidores que mantenían el fichero HOSTS.TXT antes de que se diseñara el DNS.

Sin embargo, aún se puede mejorar algo más. La información obtenida en una búsqueda puede que se necesite después. Por ello, el servidor de nombres local la guardará en una caché local. Así, cuando volvamos a preguntar por un nodo de groucho.edu, el servidor local ya podrá dirigirse directamente al servidor de nombres de esa zona sin pasar por los servidores raíz. Por supuesto, el servidor de nombres no puede mantener la caché eternamente, sino descartarla cada cierto tiempo. Este tiempo de expiración se conoce como TTL o tiempo de vida. En la base de datos del DNS se especifica este parámetro.

5.3.2 Servidores de nombres

Cuando un servidor de nombres mantiene toda la información acerca de una zona se le llama autorizado para esa zona. Cualquier petición para esa zona será enviada a uno de esos servidores maestros.

Para tener una representación coherente de la zona, sus servidores maestros deben estar sincronizados. Para ello, a uno de ellos se le nombra servidor primario, que obtiene la información de zona a partir de unos ficheros locales, y a los demás se les nombra servidores secundarios. Estos últimos cargan la información de la zona pidiéndosela al primario cada cierto tiempo.

Las razones para que existan varios servidores autorizados por cada zona son dos: repartir la carga de trabajo y lograr tolerancia a fallos. Así, si un servidor cae, todas las peticiones se repartirán entre los demás servidores autorizados que haya. Por supuesto, esto no protege contra fallos internos o bugs del propio software DNS.

También es posible tener servidores de nombres que no mantengan información autorizada de ningún dominio. Este tipo de servidores es útil, pues al mantener una caché con los nombres que resuelven, disminuyen la carga de la red y de otros servidores.

5.3.3 La base de datos DNS

En las bases de datos del DNS se mantiene más información que la necesaria para traducir nombres a direcciones IP. Dicho de otra forma, en DNS se mantienen distintos tipos de registros.

Un elemento simple de información en el DNS se conoce como registro de recurso o RR. Cada registro tiene un tipo asociado a él, describiendo qué clase de datos contiene, y una clase indicando el tipo de red al que se aplica. Se trata de adecuar a diferentes esquemas de red, aunque para direcciones IP se usa siempre la clase IN (INternet), pero hay otras como las redes Hesiod (que se usan en el MIT). El registro más habitual es el de tipo A, que relaciona un nombre totalmente cualificado con una dirección IP.

Un nodo puede admitir más de un nombre. Pero solo uno de ellos será “oficial” o canónico, mientras que los demás son alias del primero. La diferencia es que el canónico se define en un registro de tipo A, mientras que los alias se definen en registros CNAME que apuntan al nombre canónico.

Además de los registros A y CNAME, se puede ver que hay un registro especial al principio del fichero, con varias líneas. Se trata del registro SOA o de inicio de autoridad, que mantiene información general sobre el servidor de nombres. Por ejemplo, el tiempo de vida por defecto de todos los registros que mantiene.

Nótese que aquellos nombres que no finalicen en un punto serán interpretados como relativos al dominio en cuestión. El nombre especial “@” usado en el registro SOA representa al dominio completo.

Hemos visto que los servidores para el dominio groucho.edu deben tener conocimiento sobre los servidores de la zona physics para poder reenviarles las peticiones para esta. Esto se suele incluir en los registros NS que poseen el nombre de

los servidores en notación FQDN, y un registro A que da la dirección IP para ese servidor.

5.4 Los ficheros de inicio rc para trabajo en redes.

Los ficheros rc son shell scripts que se ejecutan durante el arranque del sistema para configurarlo. Son ejecutados por el proceso init, y ponen en marcha los demonios básicos como sendmail o cron y además configuran parámetros de la red como la dirección IP y el nombre del host. Estos scripts se suelen encontrar en /etc/rc.d o en /etc.

Lo que vamos a hacer aquí es describir los ficheros rc que configuran TCP/IP. En Linux son dos: rc.inet1 y rc.inet2. El primero configura parámetros básicos como direcciones IP e información de rutado. El segundo lanza los demonios TCP/IP, principalmente inetd, quien se encargará de lanzar cuando haga falta los telnetd y demás.

En muchos sistemas se juntan los dos ficheros en uno, el rc.inet o rc.net. No tiene importancia el nombre, siempre que se ejecuten en el momento adecuado durante el arranque. Para conseguirlo, init tiene que saberlo, y para ello existen entradas específicas en el fichero inittab. En el peor caso tendría usted que crear las entradas para rc.inet1 y rc.inet2 en dicho fichero.

Como hemos dicho, rc.inet1 configura los parámetros básicos de red. Esto incluye el número IP, dirección de red, y la tabla de rutado. Estas tablas se usan para rutar los datagramas entrantes y salientes de otras máquinas. Lo más simple es tener tres rutas: una para enviar paquetes a su propia máquina, otra para enviarlos a otra máquina de la subred y una tercera para enviarlos a máquinas de otras subredes (a través de una pasarela). Para configurar esto se usan los programas ifconfig y route, programas que suelen estar en /etc.

ifconfig se utiliza para configurar el dispositivo interfaz de red con los parámetros que necesita, como la dirección IP, la máscara, dirección de broadcast y otros. route, por su lado, se utiliza para crear o modificar entradas de la tabla de rutado.

Para muchas configuraciones, el siguiente rc.inet1 puede valer, aunque, por supuesto, necesitará poner sus propias direcciones IP y demás.

```
#!/bin/sh

#/etc/rc.d/rc.inet1 – Configuración de TCP/IP

# Configuración del dispositivo 'loopback'

HOSTNAME='hostname'

/etc/ifconfig lo 127.0.0.1          # utiliza por defecto la máscara 255.0.0.0

/etc/route add 127.0.0.1          # una ruta apunta al dispositivo 'loopback'

# Configuración del dispositivo ethernet. Si solo se usa el 'loopback',

# comentar las líneas siguientes.

# EDÍTELO con sus propios datos.

IPADDR="128.253.154.32"           # PONGA aquí su dirección IP

NETMASK="255.255.255.0"          # PONGA aquí su máscara de red

NETWORK="128.253.154.0"          # PONGA aquí su dirección de red

BROADCAST="128.253.154.255"       # PONGA aquí su dirección 'broadcast'

# si la tiene. Si no, elimine la línea.

GATEWAY="128.253.154.1"           # PONGA aquí su dirección de pasarela

/etc/ifconfig eth0 ${IPADDR} netmask ${NETMASK} broadcast ${BROADCAST}

# Si no tiene dirección de 'broadcast', ponga la anterior línea así:

#/etc/ifconfig eth0 ${IPADDR} NETMASK ${NETMASK}

/etc/route add ${NETWORK}

# Lo que sigue solo hace falta si hay pasarela, o sea, si su subred esta

# conectada a otra red.

/etc/route add default gw ${GATEWAY} metric 1

# Fin de la configuración de ethernet
```

Quizás tenga que estudiarse un poco más el tema para su instalación particular, aunque en la mayor parte de los casos el fichero anterior será suficiente.

rc.inet2 arranca servidores usados por TCP/IP. El más importante es inetd, que queda en segundo plano y escucha por varios puertos de la red. Cuando una máquina intenta conectarse por uno de ellos (por ejemplo, por el de telnet), inetd envía una copia del servidor correspondiente (en este caso, in.telnetd) para que controle el puerto afectado. Esto es mejor que mantener en ejecución todos los servidores de red necesarios (múltiples copias de in.telnetd, in.ftpd y demás). inetd los arranca conforme se van necesitando.

En rc.inet2 se arrancan también otros demonios. syslogd se ocupa de acumular los mensajes generados por el núcleo y diversas aplicaciones y tratarlos según diga el fichero /etc/syslogd.conf (guardarlos en ficheros, sacarlos por consola,etc.). routed se ocupa de la información de ruteado dnámico, cuando su sistema intenta enviar paquetes a otra red puede requerir nuevas entradas en las tablas de ruteado, que routed trata sin necesidad de intervención del usuario.

El ejemplo siguiente sólo arranca un número mínimo de servidores. Existen otros que pueden interesarle, como el NFS. Cuando instale TCP/IP en su sistema, es mejor empezar con una configuración sencilla y luego complicarla según sus necesidades.

Observe que en el fichero siguiente se asume que los servidores de red se encuentran en /etc, pero pueden estar en otro sitio (en /sbin, por ejemplo).

```
#!/bin/sh

# Ejemplo de /etc/rc.d/rc.inet2

# Arrancar syslogd

if [ -f /etc/syslogd ]

then

    /etc/syslogd

fi

# Arrancar inetd

if [ -f /etc/inetd ]

then

    /etc/inetd
```

```
fi
# Arrancar routed
if [ -f /etc/routed ]
then
    /etc/routed -q
fi
# Hecho!
```

Otro servidor que puede interesarle es named, servidor de nombres, que traducirá nombres (locales) a direcciones IP y viceversa. Si no hay servidor de nombres en su subred o quiere proporcionar nombres nuevos a la misma, necesitará arrancar named. Su configuración es más compleja y requiere cierto cuidado y planificación, por lo que le recomendamos consultar bibliografía específica. Sin embargo, no es habitual tener que instalar un servidor de nombres en su sistema.

5.4.1 /etc/hosts

/etc/hosts lleva una lista de direcciones IP y nombres de máquinas que les corresponden.

En general, /etc/hosts solo contiene entradas para su máquina y quizás alguna otra “importante”, como servidores de nombres o pasarelas. Su servidor de nombres local proporciona a otras máquinas traducción automática del nombre de su host a su dirección IP.

Por ejemplo, si su máquina es frsf.utn.com con la dirección IP 128.253.154.32, su /etc/hosts sería como este:

```
127.0.0.1    localhost
128.253.154.32    frsf.utn.com    frsf.
```

Si solo usa el “loopback”, la única línea necesaria es la que tiene el número 127.0.0.1, añadiendo tras local host el nombre de su máquina.

5.4.2 /etc/networks.

El fichero /etc/networks tiene direcciones de su red y otras, y es usado por el comando route. Permite dar nombre a las redes.

Cada subred que quiera añadir a route debe aparecer en /etc/networks.

Por ejemplo:

```
default 0.0.0.0      # rutado por defecto - obligatorio
loopnet 127.0.0.0 # red de 'loopback' - obligatorio
mynet 128.253.154.0 # Ponga aquí su dirección de red
```

5.4.3 /etc/host.conf

Este fichero dice a su sistema cómo resolver los nombres de los hosts. Debe contener dos líneas:

```
order hosts,bind
multi on
```

Estas líneas indican a los mecanismos de resolución que empiecen buscando en el fichero /etc/hosts y luego pregunten al servidor de nombres, si existe. La entrada multi permite que para un nombre de máquina haya varias direcciones IP en /etc/hosts.

5.4.4 /etc/resolv.conf

En este fichero se configura el mecanismo de resolución, especificando la dirección del servidor de nombres y el nombre del dominio de su máquina. El dominio es como un nombre de host "mutilado".

Por ejemplo, si su máquina se llama frsf.utn.com, el dominio será utn.com.

Como fichero /etc/resolv.conf de ejemplo, veremos el caso de la máquina wan.grupo.com cuyo servidor de nombres es el 127.253.154.5:

```
domain      grupo.com
nameserver  127.253.154.5
```

Con líneas nameserver adicionales podrá especificar la existencia de varios servidores de nombres.

5.4.5 Ajuste del nombre de su host.

Para activar el nombre de su host debe usar el comando hostname. Esto suele hacerse en un fichero como /etc/rc.local. Busque en sus ficheros rc y busque una llamada a hostname como la siguiente:

```
/bin/hostname frsf.utn.com
```

Vea que hay que especificar el nombre completo (dominio incluido).

5.4.6 Problemas con la configuración

Una vez que haya preparado los ficheros anteriores, habrá que reiniciar Linux para que reconozca las nuevas configuraciones. Luego tendrá que hacer pruebas, para las que lo más indicado es probar aspectos individuales de la red y no tratar de empezar, por ejemplo, lanzando un proceso Mosaiccon a una conexión X.

Con el comando netstat puede ver las tablas de rutado. Esta suele ser la principal fuente de problemas. En la página del manual para este comando encontrará la sintaxis adecuada. Para comprobar que funciona su conexión, le sugerimos probar un cliente como telnet para ver si puede conectarse a máquinas de su subred y de otras redes. Esto puede ponerle sobre la pista del problema.

Por ejemplo, si puede conectarse a máquinas de otras subredes pero no de la suya propia, puede tratarse de un problema con la máscara de red o las tablas de rutado. Ejecutando route como root podrá jugar directamente con las entradas de la tabla.

Para hacer estas pruebas de conectividad, utilice direcciones IP y no nombres. Así, si tiene problemas para ejecutar

```
$telnet shoop.utn.com
```

la causa puede ser una configuración incorrecta del servidor de nombres. Si funciona usando la dirección IP, se puede casi asegurar que el resto de la

configuración está bien hecha. Solo falta que funcione bien el servicio de nombres (probablemente haya que especificar correctamente la dirección del servidor de nombres).

La depuración de configuraciones de red puede ser tarea difícil, y no podemos tratarla aquí. Le sugerimos, si no consigue otra ayuda, que consulte el libro *Linux Network Administrators' Guide* citado en la bibliografía del trabajo.

5.5 La Librería de Sockets

La interface de programación más común en el mundo UNIX es la Librería de Socket de Berkeley, Berkeley Socket Library. Su nombre proviene de una analogía popular que ve los puertos como sockets, y conectarse a un puerto como enchufarse. Proporciona la llamada (bind) para especificar un nodo remoto, un protocolo de transporte, y un servicio al que un programa pueda conectarse o escuchar (usando connect, listen, y accept). La librería de socket, sin embargo, es algo más general, en que proporciona no sólo una clase de sockets basados en TCP/IP (los sockets AF_INET), sino también una clase que maneja conexiones locales a la máquina (la clase AF_UNIX). Algunas implementaciones pueden manejar también otras clases, como el protocolo XNS (Xerox Networking System), o X.25.

En Linux, la librería de socket es parte de la librería C estándar libc. Actualmente sólo soporta los sockets AF_INET y AF_UNIX, pero se hacen esfuerzos para incorporar el soporte de los protocolos de red de Novell, de modo que se añadirían eventualmente una o más clases de sockets.

6. APÉNDICE A. EL PROYECTO

Arribamos ya al final de nuestra cita, es el momento de aunar todos los conocimientos adquiridos a lo largo del trabajo y pensar como verdaderos *Administradores de Redes*.

Mi propuesta, si bien es muy amplia, permite integrar en condiciones homogéneas los conceptos de redes. Para lograr así un sistema de biblioteca seguro, con excelentes conexiones sobre plataformas estables y conocidas.

Mi propuesta se basa en la tranquilidad del usuario, en la tranquilidad del administrador, y por ende, la comodidad de ambos.

Lamentablemente, no tengo conocimientos técnicos sobre el estado de las redes y su distribución actual en las distintas regionales adheridas a Biblioteca 2000. Excepto en la regional Santa Fe, a la cuál asisto regularmente.

Para que mi trabajo se pueda adaptar a toda la Universidad Tecnológica Nacional y no se limite en lo que respecta a tiempos e integración con otras universidades del mundo, es de extrema necesidad lograr una muy buena abstracción de requisitos. En otras palabras, como no todas las regionales tienen el mismo tipo de red, o en algunos casos, algunas no tienen interconectadas sus computadoras, sería un error tratar sobre una determinada topología y hablar sólo de algún tipo especial de protocolo. Por este motivo, voy a ser lo más genérico posible en el tema, con la idea de que así, los distintos aspectos tratados podrán adaptarse con facilidad a las condiciones locales.

Las redes de conmutación de paquetes que deben recorrer distancias geográficas grandes, son fundamentalmente diferentes de las que recorren distancias cortas. Para ayudar a caracterizar las diferencias en la capacidad y las proyecciones de uso, la tecnología de conmutación de paquetes se divide con frecuencia en dos grandes categorías: *Wide Area Networks (redes de área amplia o WAN)* y *Local Area Networks (redes de área local o LAN)*.

Muchas tecnologías de tipo WAN no tienen un límite de distancia para su recorrido, esto hace que estén definidas para cubrir a todo el mundo, y en algunos

casos distancias aún mayores. Por lo común, las WAN operan más lentamente que las LAN, y tienen tiempos de retraso mayores entre sus conexiones. La velocidad normal para una WAN llega a un rango que va desde los 56Kpbs a 155Mbps. Los retardos para una WAN pueden variar de unos cuantos milisegundos a varias decenas de segundos.

Las tecnologías LAN proporcionan las velocidades de conexión más altas entre computadoras, pero sacrifican la capacidad de recorrer largas distancias. Una LAN opera a una velocidad que va desde los 10Mbps a los 2Gbps. Debido a que la tecnología LAN cubre distancias cortas, ofrece tiempos de retraso mucho menores que las WAN. Los tiempos de retraso de una LAN pueden ser cortos, como un milisegundo; o largos, como unos 10 milisegundos.

Podemos clasificar a las redes de tipo LAN en 4 topologías básicas diferentes: topología en estrella, en bus, en anillo y por último una híbrida.

Las cuales hemos visto, sin mucho detalle, durante el desarrollo del trabajo.

En la topología de tipo bus, el servidor y todos los equipos terminales están conectados a una línea compartida (comúnmente un cable coaxial) que mantiene a todas las computadoras interconectadas. El problema con este tipo de distribuciones es que para añadir una terminal hay que cortar obligatoriamente el flujo de transmisión de datos.

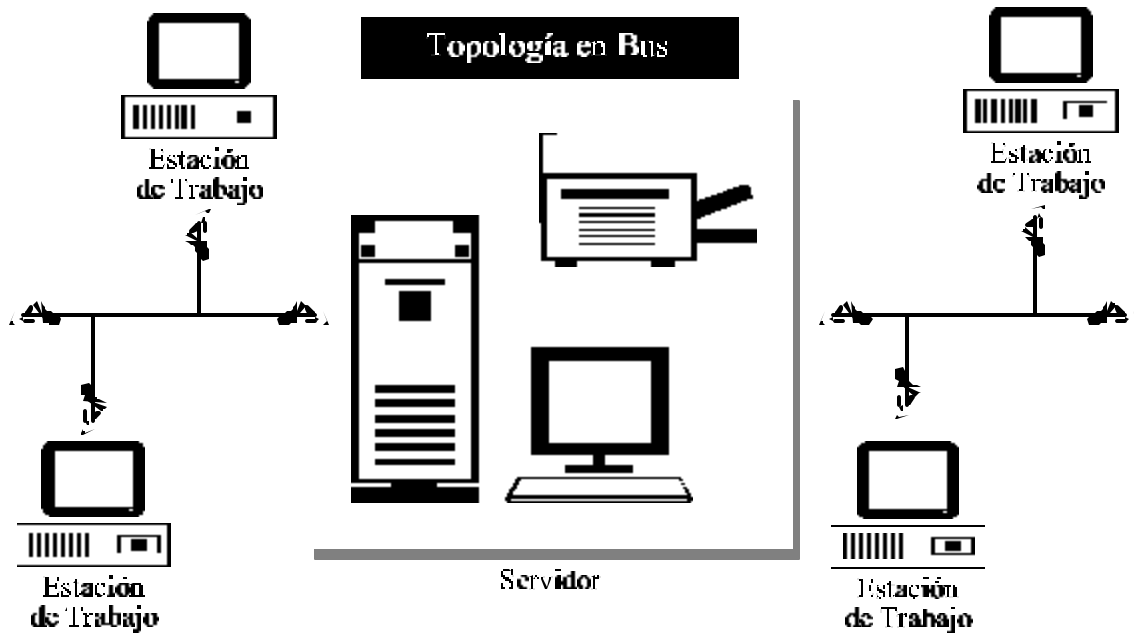


Ilustración 6-1 (Topología en bus)

En la topología estrella, el servidor posee una salida denominada multipuerto, y cada terminal está conectada directa y físicamente al servidor.

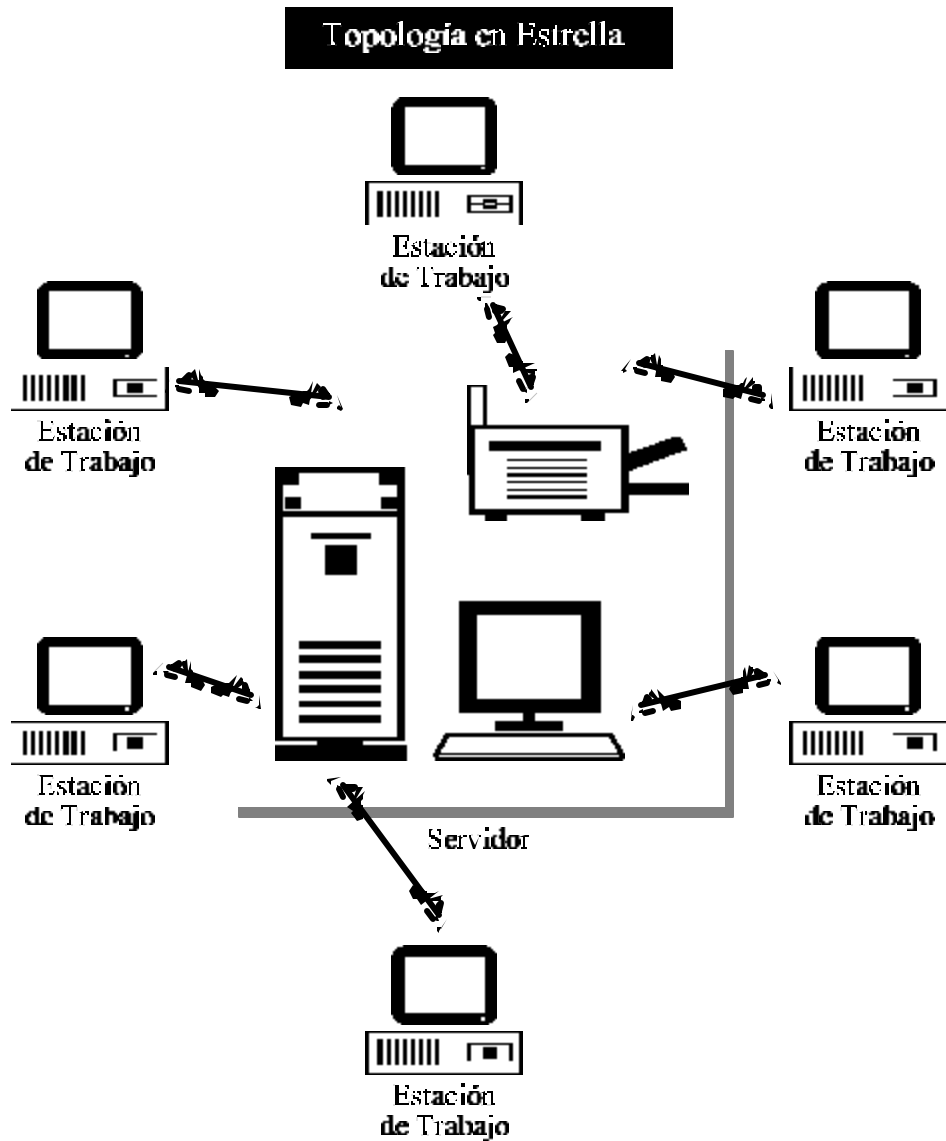


Ilustración 6-2 (Topología en estrella)

La topología en Anillo, es muy similar a la de bus, radican en la diferencia que la de Anillo tiene su último y primer nodo unidos, formando así un círculo de comunicación.

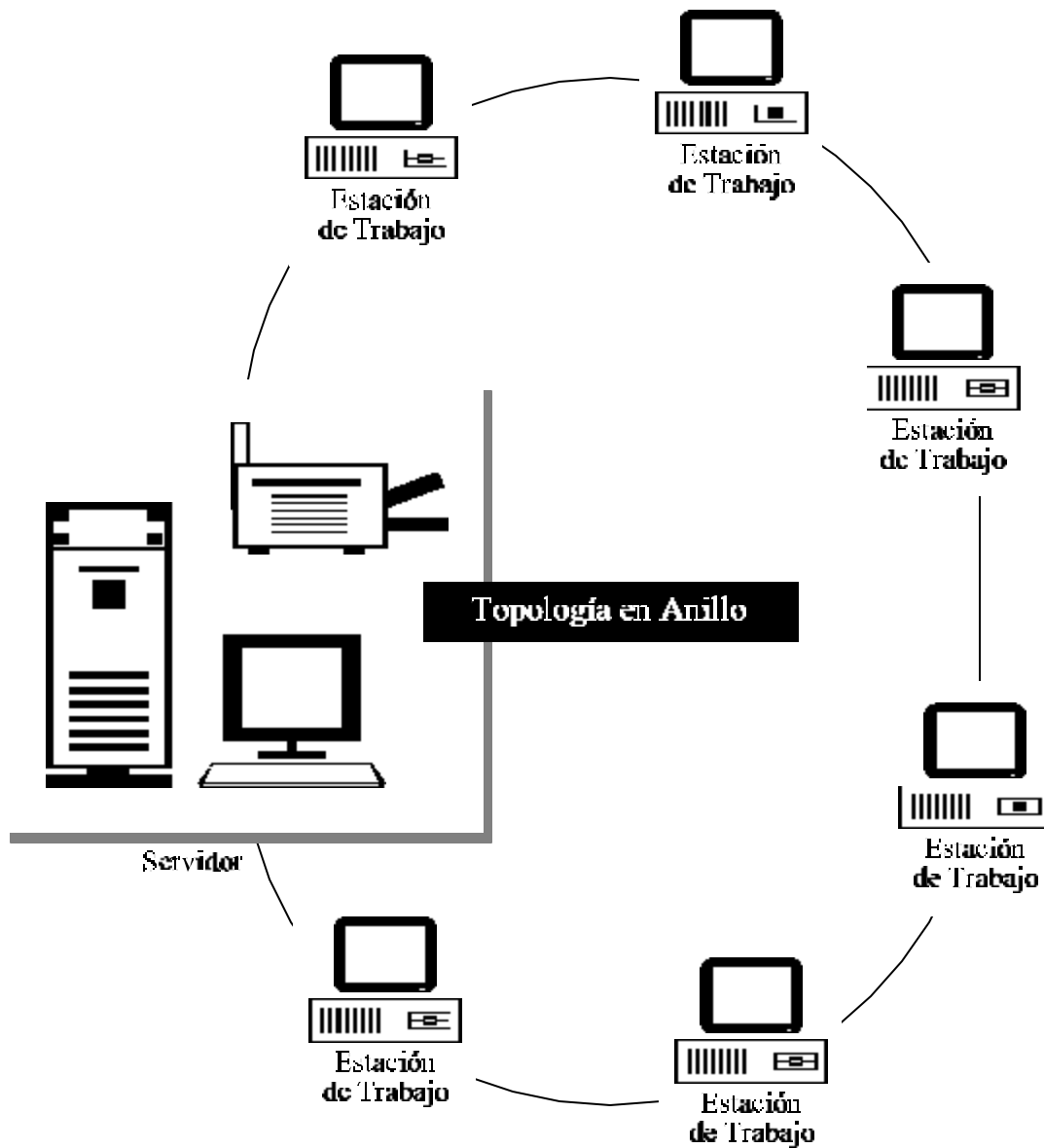


Ilustración 6-3 (Topología en anillo)

Ahora que hemos revisado rápidamente las diferentes topologías de conexión disponibles para redes LAN, volvamos a nuestro proyecto de biblioteca.

En nuestro sistema, serán necesarias redes de tipo WAN y LAN, las cuales ya fueron descritas.

Con respecto a las WAN, sabemos de la existencia de una red suprema a nivel mundial que es Internet. Si evaluamos nuestros recursos, utilizarla será la manera más eficiente, económica y segura.

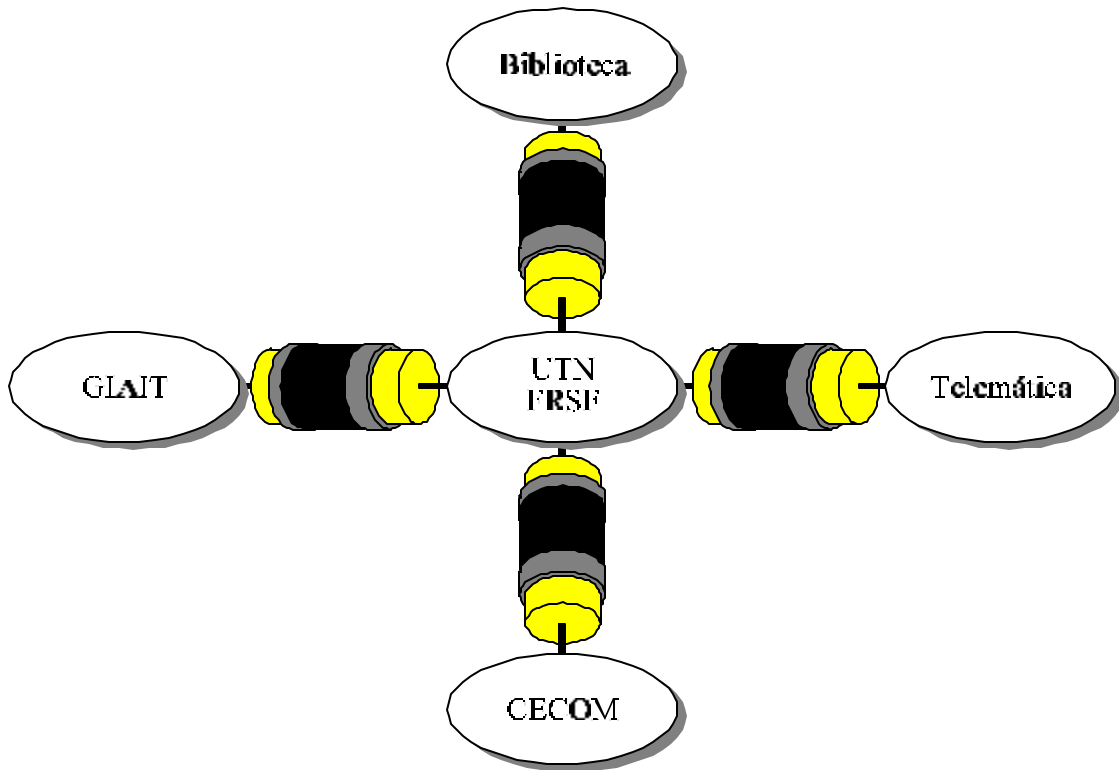


Ilustración 6-4 (Modelo de red interna a la facultad)

Al adoptar la arquitectura interna de una internet, estaremos creando un intranet (concepto que se detalla a continuación). Sugiero entonces, como propuesta de trabajo, el desarrollo de redes LANs para las diferentes facultades y grupos dentro de las mismas, enlazadas entre sí por medio de una intranet que utilizará los recursos físicos de Internet para mantener una buena comunicación.

En el transcurso del trabajo, se ha visto en detalle o a gran escala (con dependencia sobre el tema), como debe administrarse una red en Linux. Sabemos entonces, mínimamente, poner en funcionamiento el concepto de una internet. Básicamente, ésta internet hará las veces de Intranet local a la regional (o facultad). A un nivel global, utilizaremos nuevamente el concepto de intranet; esta vez, nuestra intranet estará literalmente dentro de Internet.

De esta manera, será totalmente innecesario realizar conexiones físicas con otras facultades o regionales, ya que simplemente se utilizarán los recursos de Internet para comunicar las peticiones externas de los usuarios.

En la ilustración 6-4, cada grupo, ya sea biblioteca, GIAIT, etc. tiene dispuesta una red interna interconectada de la manera más conveniente a sus recursos locales.

Hablamos de intranets, ¿pero cuál es exactamente el significado de ésta palabra? ¿como podemos adaptarla a nuestro sistema de Biblioteca 2000?

6.1 ¿Qué es intranet?

Intranet es la denominación que se les da a las redes corporativas basadas en TCP/IP y que trabajan con aplicaciones Internet. Este tipo de red puede estar conectada a Internet a través de un router limitando el acceso desde el exterior mediante el establecimiento de Firewalls. También se puede optar por aislar la red intranet de exterior con lo cual sólo se podrá acceder desde las instalaciones de la organización para la que trabaje la red.

Si reflexionamos por un momento nos podemos percatar que Internet nació a partir de la unión de una multitud de redes intranet, a partir de dicha unión se comenzaron a crear protocolos de comunicación universales como el Gopher, WWW o los distintos protocolos de correo electrónico. Se produce ahora un fenómeno de retroalimentación en el que las distintas aplicaciones desarrolladas para un entorno amplio como el de Internet comienzan a operar en entornos corporativos más o menos reducidos.

6.2 Intranet vs. Internet

Hasta el momento he presentado a intranet como un concepto opuesto a Internet. Sin embargo la realidad es bien distinta ya que son dos conceptos complementarios de una misma realidad.

Internet está orientado a la comunicación global, a la búsqueda y recuperación de información para el gran público. Por su parte intranet se centra en los flujos de trabajo

(Workflow), en el desarrollo cooperativo de tareas y en el desarrollo e instalación de aplicaciones de forma modular y centralizada.

No obstante estas dos estructuras informativas no deben desarrollarse de forma paralela, sino que deben coincidir en algunos momentos. Ser precisamente en los puntos donde Internet e intranet coincidan donde se producir un mayor desarrollo para ambos tipos de redes.

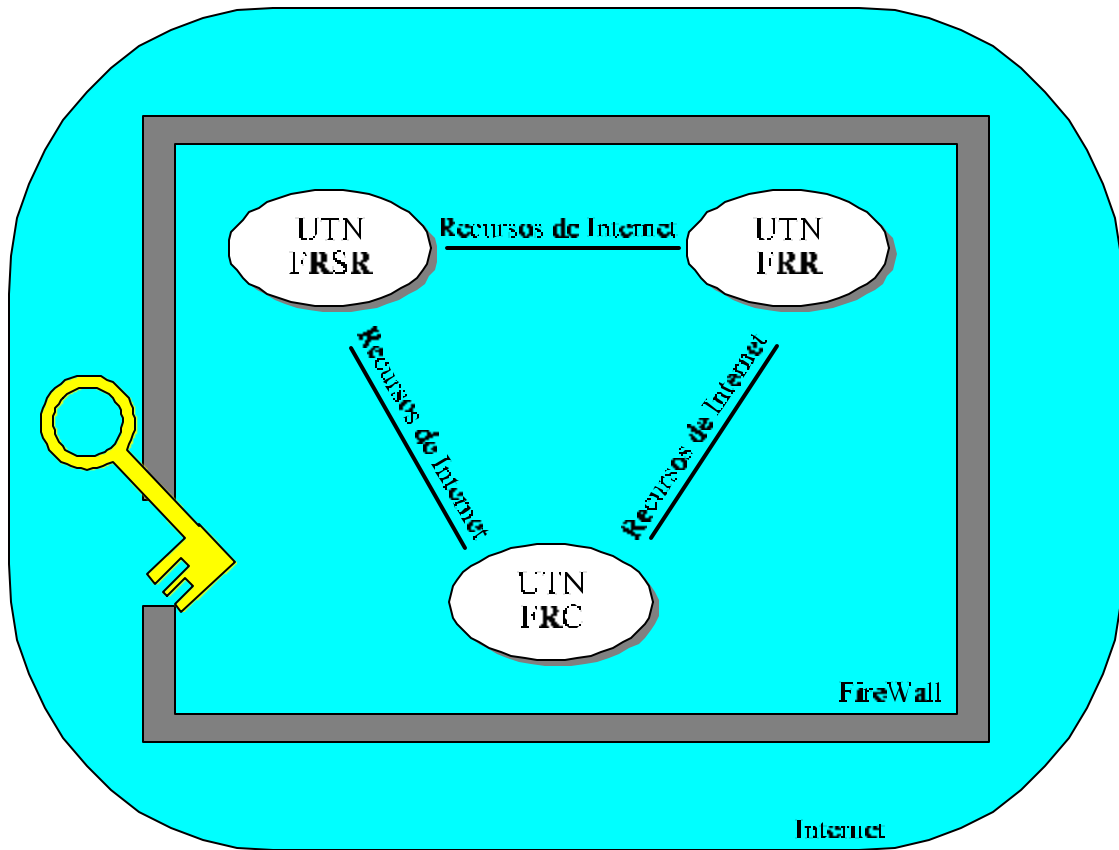


Ilustración 6-5 (Modelo de intranet Biblioteca 2000)

6.3 El papel de Linux en intranet

Cuando se habla de Internet o intranet hay que abordar aspectos más pragmáticos (no hay que olvidar que se esté hablando de una realidad). Uno de estos aspectos es la adopción de un Sistema operativo para intranet, y Linux tiene mucho que decir al respecto.

Una intranet es una red muy dinámica con cambios continuos en su estructura y funcionalidad, esto implica el disponer de un Sistema operativo flexible y con la posibilidad de ser ajustado a medida. El carácter de libre distribución de Linux y el disponer de su código fuente permite realizar una serie de cambios destinados a mejorar su rendimiento dentro de una organización.

Un sistema operativo Linux puede constituir un núcleo que brinde soporte a Servidores Web, Gopher, FTP y de correo electrónico. Además, la inclusión en las distribuciones de este Sistema operativo de lenguajes de programación permite el desarrollo de nuevas aplicaciones y protocolos.

Por otra parte, Linux permite la comunicación entre los miembros de una organización a través de correo electrónico e integrar el almacenamiento de documentos electrónicos en servidores FTP o Gopher.

Incluso pueden aprovecharse protocolos ya establecidos para el desarrollo de aplicaciones propias con el Interfaz común de un programa cliente. El uso de estas aplicaciones por parte del usuario final es independiente de la máquina con la que trabaje o del sistema operativo que utilice, solamente tendrá que tener instalado su cliente WWW.

Este aspecto es muy importante, ya que con un servidor Linux, no es necesario que los equipos terminales estén ejecutando un mismo sistema operativo, facilitando así, problemas de compatibilidad. También es bueno recalcar que no todos los usuarios estarían obligados a aprender Linux, cada uno (dentro de sus posibilidades) puede “entrar a la red” con solo poseer un cliente WWW y un sistema operativo que permita su ejecución.

6.4 Usuarios y formación

Hay que tener en cuenta que existen varios niveles dentro una intranet, los cuales se corresponden al perfil de usuario:

1. Gestor de Sistemas Informáticos. Se encargaría de diseñar la estructura informática de la organización, planificar y realizar instalaciones de sistemas operativos y aplicaciones muy específicas, programar aplicaciones, etc.
2. Gestor de Información. Sus funciones serían la de crear el Sistema de Información, diseñar y mantener la estructura de Bases de Datos y servidores de Información, diseñar aplicaciones para la gestión de información, etc.
3. Usuario final. Utiliza los recursos informáticos e informativos para el desarrollo de su trabajo, en conjunción con el resto de miembros de la organización. Para el proyecto de Biblioteca 2000, existen (a grandes rasgos) dos tipos de usuarios finales.
 - por un lado están las empresas, las cuales tendrían una cuenta en el servidor donde se conecten normalmente y allí podrían guardar la información que les interese.
 - por otro, los usuarios normales, que bien podrían ser estudiantes situados físicamente en la facultad o simplemente desde su casa por medio de una línea telefónica y una conexión SLIP.

7. APÉNDICE B. LA LICENCIA GENERAL PÚBLICA. GNU

Versión 2, junio de 1.991

Copyright 1.989, 1991 Free Software Foundation,
Inc. 675 Mass Ave, Cambridge, MA02139 USA

Se permite a todo el mundo copiar y distribuir copias idénticas de este documento de Licencia, aunque no se permite su modificación.

7.1 Preámbulo

Las Licencia de uso de casi todo el software han sido establecidas para quitarle la libertad de compartirlo y modificarlo. En contraste con esta costumbre, la Licencia General Pública GNU pretende garantizar su libertad de compartir y modificar software gratuito, a fin de asegurar que el software es gratuito para todos sus usuarios. Esta Licencia General Pública es de aplicación para la mayor parte del software de la Free Software Foundation y para cualquier otro Programa cuyos autores se comprometen a su utilización. (Otro Software de la Free Software Foundation está cubierto alternativamente por la Licencia Pública de biblioteca GNU). También puede aplicar ésta a sus programas.

Cuando hablamos de Software gratuito nos referimos a la libertad, no al precio. Nuestras Licencias Generales Públicas han sido diseñadas para asegurarle la libertad necesaria para distribuir copias de software gratuito (y de cobrar por este servicio, si así lo desea), que reciba código fuente o pueda obtenerlo si lo desea, que pueda modificar el software o utilizar partes del mismo en nuevos programas gratuitos y que sepa que puede hacer estas cosas.

A fin de proteger sus derechos, es necesario constituir restricciones que impidan que alguien pueda denegarle estos derechos o pedirle que renuncie a los mismos. Estas restricciones implican también ciertas responsabilidades si distribuye copias del software o lo modifica.

Por ejemplo, si distribuye copias de este Programa, tanto gratis o con ánimo de lucro, deberá conceder a los recipientes del mismo todos los derechos con los que cuente. Deberá asegurarse de que todos los recipientes reciben o pueden obtener el código fuente. Y deberá asimismo mostrarle estos términos, a fin que puedan conocer sus derechos.

Protegemos sus derechos con dos pasos: (1) Copyright del software y (2) le ofrecemos esta Licencia que le concede permiso legal para copiar, distribuir y/o modificar el software.

Además, para la protección de cada autor así como la nuestra propia, deseamos asegurarnos de que todo el mundo comprende que este software gratuito no cuenta con garantía alguna. Si el software es modificado por alguna otra persona y posteriormente distribuido, deseamos que sus destinatarios sepan que lo que reciben no es el producto original, de forma que cualquier problema introducido por otras partes no pueda reflejarse en la reputación profesional de los autores originales.

Finalmente, cualquier Programa gratuito se ve constantemente amenazado por patentes de software. Deseamos evitar el peligro de que los redistribuidores de un Programa gratuito obtengan licencias de patente a título individual, lo que haría de éste un Programa privado.

A fin de impedir esto hemos dejado muy claro que cualquier patente obtenida deberá ser licenciada para la utilización gratuita de todos o no deberá ser licenciada en absoluto.

7.2 Términos y condiciones de la licencia general pública de GNU para la realización de copias, distribución y modificación

1. Esta Licencia es de aplicación a cualquier Programa u otro trabajo que contenga un aviso incluido por el titular de los derechos de copyright estableciendo su libertad de distribución bajo los términos de esta Licencia General Pública. El "Programa", a continuación, hace referencia a cualquier Programa o trabajo y un "trabajo basado en el Programa" indica el Programa

o cualquier tipo de trabajo derivado bajo las leyes de Copyright: es decir, un trabajo que contenga el Programa o una porción del mismo, bien en versión exacta o bien con modificaciones y/o traducido a otro idioma. (En adelante, las traducciones quedan incluidas, sin limitación alguna, en el término “modificación”). Cada beneficiario de esta Licencia quedará indicado por el término “usted”. Las actividades que no incluyan copia, distribución y modificación no quedarán protegidas por esta Licencia; quedan fuera de su alcance. El acto de ejecución del Programa no queda limitado y los resultados obtenidos de la ejecución del Programa (independientemente de que haya sido obtenido mediante la ejecución del Programa). La validez de esto dependerá de lo que haga el Programa.

2. Podrá copiar y distribuir copias exactas del código fuente del Programa según lo hubiera recibido, utilizando cualquier medio, siempre que publique, en cada copia, un aviso adecuado de derechos de copyright y relativo a la ausencia de garantía alguna; y de que proporcione a todos los destinatarios del Programa una copia de esta Licencia junto con el mismo. Podrá cobrar honorarios por el acto físico de transferencia de una copia y podrá ofrecer servicios de protección de garantía a cambio de honorarios.
3. Puede modificar su copia o copias del Programa o cualquier porción del mismo y podrá asimismo copiar y distribuir dichas modificaciones o trabajos, siempre que cumplan las condiciones adicionales que se detallan a continuación:
 - Deberá asegurarse que los archivos modificados incorporen avisos prominentes que establezcan que ha modificado los archivos, así como la fecha de dicha modificación.
 - Deberá asegurarse de que cualquier trabajo que distribuya o publique, que pudiera en su totalidad o en parte, contener o haber sido derivado del Programa o de cualquier parte del mismo, sea licenciado en su totalidad y sin cargo alguno a terceros que lo requieran bajo los términos de esta Licencia.

- Si el Programa modificado lee normalmente comandos interactivamente cuando se ejecuta, deberá asegurarse de que, cuando se ponga en marcha su ejecución para dicha utilización interactiva de la forma más común, imprima o muestre un proclama que incluya un aviso correspondiente relativo a los derechos de copyright y una clarificación de la falta de garantía (o estableciendo la provisión de dicha garantía por su parte) y que confirme que los usuarios pueden redistribuir el Programa bajo estas condiciones, e informe al usuario cómo visualizar una copia de esta Licencia. (Excepción: si el Programa es de por sí interactivo pero no imprime normalmente dicha proclama, tampoco se requerirá que su trabajo basado en el Programa deba imprimirla).
 - Estos requisitos son de aplicación al trabajo modificado en su totalidad. Si hubiera secciones identificables de dicho trabajo no derivadas del Programa y que pudieran considerarse trabajos razonablemente independientes, entonces esta Licencia, así como sus términos, no serían de aplicación a dichas secciones cuando las distribuya como trabajos independientes. Pero cuando distribuya las mismas secciones como parte de un trabajo completo mayor basado en el Programa, la distribución del trabajo completo deberá realizarse bajo términos de esta Licencia, cuyos permisos con respecto a otras licencias se ampliarán a la totalidad del trabajo completo y por lo tanto a cada una de las partes del trabajo irrespectivamente de quién lo ha escrito. Por todo esto, no es el propósito de esta sección reclamar derechos o disputar sus derechos a trabajos escritos enteramente por el titular de esta Licencia sino ejercitar el derecho a controlar la distribución de trabajos derivados o colectivos basado en el Programa. Además, la simple agregación de otro trabajo no basado en el Programa a éste (o con un trabajo basado en el Programa) en un volumen de un medio de almacenamiento o distribución no somete al otro trabajo al alcance de esta Licencia.
4. Podrá copiar o distribuir el Programa en código objeto o formato ejecutable, siempre que observe asimismo una de las condiciones siguientes:

- Lo acompañe con el correspondiente código completo leíble por el sistema, que deberá distribuirse en un medio normalmente utilizado para intercambios de software; o
- Lo acompañe con una oferta por escrito, de facilitar terceros, a un precio no superior al coste de realizar la distribución física del código, una copia completa leíble por la máquina del correspondiente código fuente, a distribuir bajo los términos reseñado en un medio normalmente utilizado para intercambios de software; o
- Lo acompañe con la información que haya recibido en relación a la oferta de distribución del código fuente.
- El código fuente de un trabajo significa la forma preferida del trabajo para la realización de modificaciones al mismo. Para un trabajo ejecutable, el código fuente completo significa todo el código fuente para todo los módulos que contiene, más cualesquiera archivos de definición de interfaz asociados, más las secuencias que se utilicen para controlar la compilación y ejecución del ejecutable. Sin embargo, como una excepción especial, el código especial distribuido no necesita incluir nada que sea normalmente distribuido con los componentes principales del sistema operativo en el que se ejecuta el ejecutable, a menos que el propio componente acompañe al ejecutable. Si llegase a realizarse una distribución de código objeto o ejecutable ofreciendo acceso a copiar de un lugar designado, el ofrecimiento de acceso equivalente a copiar el código fuente del mismo lugar contará como una distribución del código fuente, incluso cuando las terceras partes no tengan que copiar el código fuente conjuntamente con el código objeto.

5. No podrá copiar, modificar, sublicenciar o distribuir el Programa salvo en la forma en que esta Licencia expresamente lo permita. Cualquier otro intento de copia, modificación, sublicencia, o distribución del Programa será nula y dará fin de forma automática a sus derechos bajo los términos de esta

Licencia. Sin embargo, las licencias correspondientes a terceros que hubieran podido recibir copias, o derechos del titular de esta Licencia bajo los términos de la misma no quedarán canceladas siempre que dichas partes observen sus condiciones.

6. No tiene obligación alguna de aceptar esta Licencia, ya que no ha firmado la misma. Sin embargo, nada aquí le concede permiso de modificación o distribución del Programa o sus trabajos derivados. Estas acciones quedarán prohibidas por la ley si decidiera no aceptar esta Licencia. Por lo tanto, mediante la modificación o distribución del Programa (o de cualquier trabajo basado en el mismo) indicará su aceptación de esta Licencia y de sus términos y condiciones para la copia, distribución o modificación del Programa o de los trabajos basados en el mismo.
7. Cada vez que redistribuye el Programa (o cualquier trabajo basado en el mismo), el destinatario recibe automáticamente una Licencia del titular original para copiar, distribuir o modificar el Programa en base a estos términos y condiciones. No podrá imponer restricción adicional alguna sobre el ejercicio, por parte del destinatario, de los derechos que aquí se conceden. No tendrá responsabilidad alguna en lo que atañe a asegurar la observación, por terceros, de las condiciones a la que está sujeta esta Licencia.
8. Si, como consecuencia de una sentencia judicial de infracción de derechos de patente o por cualquier otra razón (no limitada a cuestiones de patente), se le impusiera condición alguna (por sentencia judicial, acuerdo o cualquier otra vía) que estuviera en contradicción con las condiciones de esta Licencia, dicha condición no podrá constituir excusa en lo que atañe a la obligación de observar las condiciones de esta Licencia. Si no pudiera efectuar distribuciones de forma que satisfaga simultáneamente sus obligaciones bajo los términos de esta Licencia y cualesquiera otras obligaciones que le pudieran corresponder, la consecuencia de dichas circunstancias es que no podrá distribuir el Programa en absoluto. Si cualquier porción de esta sección llegara a ser declarada inválida o no ejecutable bajo cualquier circunstancia determinada, el resto de la sección seguirá siendo de aplicación y la sección

completa seguirá siendo así mismo aplicable siempre que ocurran otras circunstancias.

9. Si la distribución y/o utilización del Programa quedara restringida en ciertos países bajo el efecto del derecho de patentes, o interfaces sujetas a copyright, el titular original de los derechos de copyright que ampara al Programa bajo esta Licencia podrá añadir una limitación explícita del ámbito de distribución geográfica que excluya a dichos países, de forma que la distribución quede permitida sólo en países o entre países que no estén sujetos a dicha exclusión. En este caso, esta Licencia incorporará la limitación de la misma forma que si hubiera sido incluida en el texto de esta Licencia.
10. La Free Software Foundation podrá publicar versiones revisadas y/o nuevas de GNU de forma periódica. Estas nuevas versiones serán similares en espíritu a la versión actual, aunque podrían diferir en detalles a fin de resolver nuevos problemas o cuestiones. Cada versión recibirá un número de versión distinto. Si el Programa especifica un número de esta Licencia que sea de aplicación a la misma y a cualquier otra posterior, tendrá la opción de seguir los términos o condiciones de dicha versión o de cualquier versión publicada en cualquier momento por la Free Software Foundation.
11. Si deseara incorporar partes del Programa a otros programas gratuitos cuyas condiciones de distribución son distintas, escriba al autor y solicite permiso. Escriba a la Free Software Foundation con respecto al software cuyos derechos de copyright corresponda a esta organización; algunas veces concedemos excepciones a estas normas. Nuestra decisión será guiada por los dos objetivos de conservar el carácter gratuito de los derivados de nuestro software gratuito y de promover generalmente la participación en el software y su reutilización.
12. INEXISTENCIA DE GARANTÍA. DEBIDO A QUE EL PROGRAMA SE LICENCIA LIBRE DE COSTE ALGUNO, EN LO QUE PERMITE LA LEY, ESTE PROGRAMA NO ESTÁ PROTEGIDO POR GARANTÍA ALGUNA,

SALVO CUANDO SE ESTABLEZCA LO CONTRARIO POR ESCRITO, LOS TITULARES DE LOS DERECHOS DE COPYRIGHT Y/O OTRAS PARTES SUMINISTRAN EL PROGRAMA "TAL Y COMO ES", SIN GARANTÍA DE NINGÚN TIPO, EXPRESA O IMPLÍCITA, INCLUYENDO, PERO SIN QUE ESTO SIRVA DE LIMITACIÓN, LAS GARANTÍAS IMPLÍCITAS DE COMERCIALIZACIÓN O UTILIDAD PARA ALGÚN PROPÓSITO DETERMINADO. LA TOTALIDAD DEL RIESGO EN RELACIÓN A LA CALIDAD Y RENDIMIENTO DEL PROGRAMA DESCANSA EN EL USUARIO. SI SE LLEGARA A ENCONTRAR QUE EL PROGRAMA ADOLECE DE ALGÚN DEFECTO, EL USUARIO DEBERÁ ASUMIR TODOS LOS COSTES QUE RESULTARAN NECESARIOS PARA SU SERVICIO, REPARACIÓN O CORRECCIÓN.

13. BAJO NINGUNA CIRCUNSTANCIA, A MENOS QUE LO REQUIERA ALGUNA LEY APLICABLE O QUE SE ACUERDE POR ESCRITO, EL TITULAR DE LOS DERECHOS DE COPYRIGHT O CUALQUIER OTRA PARTE QUE HUBIERA PODIDO MODIFICAR Y/O REDISTRIBUIR EL PROGRAMA DE ACUERDO CON LAS CONDICIONES ANTERIORMENTE RESEÑADAS, SERÁ RESPONSABLE ANTE EL USUARIO POR DAÑOS Y PERJUICIOS, INCLUYENDO DAÑOS Y PERJUICIOS GENERALES, ESPECIALES, INCIDENTALES, O CONSECUENTES, RESULTANTES DE LA UTILIZACIÓN O INCAPACIDAD DE USO DEL PROGRAMA (INCLUYENDO, PERO SIN QUE ESTO CONSTITUYA UNA LIMITACIÓN, LA PÉRDIDA DE DATOS O LA INEXACTITUD DE DICHOS DATOS, O LAS PÉRDIDAS SOSTENIDAS POR EL USUARIO O TERCEROS O UN FALLO DEL PROGRAMA CON RESPECTO A SU FUNCIONAMIENTO CON OTROS PROGRAMAS), INCLUSO SI DICHO TITULAR U OTRA PARTE HUBIERA SIDO INFORMADO CON RESPECTO A LA POSIBILIDAD DE DICHOS DAÑOS Y PERJUICIOS.

8. BIBLIOGRAFÍA

- Utilizando Linux, Tackett & Gunter. Prentice Hall
- Redes globales de información con Internet y TCP/IP, Douglas E. Comer. Prentice Hall
- Sistemas Operativos Modernos, Andrew S. Tanenbaum. Prentice Hall
- Revista Compu Magazine. Edición especial de Intranets
- Apuntes de la cátedra de Comunicaciones
- Correo electrónico de Fidonet o Internet
- Servicio de FTP de internet <http://www.infor.es/LINUX>
- Servicio de información de la Universidad de Murcia
- Teleinformática Aplicada tomo 1, Antonio Ricardo Castro Lechtaler. McGraw Hill
- TCP/IP Network Administration, Craig Hunt
- Linux Instalación y Primeros Pasos, Matt Welsh (Traducción: Proyecto LuCAS)
- Guía de Administración de Redes con Linux, Olaf Kirch (Traducción Proyecto LuCAS)

Trabajo realizado por:

Federico Gonzalez

fegonzal@telebot.com