



**UNIVERSIDAD NACIONAL AUTONOMA
DE MEXICO**

**FACULTAD DE ESTUDIOS SUPERIORES
CUAUTITLAN**

**REDES DE COMPUTADORAS. INSTALACION Y
CONFIGURACION DE UN SERVIDOR DE CORREO
ELECTRONICO BAJO EL SISTEMA OPERATIVO
LINUX**

**TRABAJO DE SEMINARIO
QUE PARA OBTENER EL TITULO DE:
LICENCIADO EN INFORMATICA
P R E S E N T A :
G O N Z A L O C E J A L E O N**

ASESOR: ING. JESUS MOISES HERNANDEZ DUARTE

CUAUTITLAN IZCALLI, EDO DE MEX.

1997.

**TESIS CON
FALLA DE ORIGEN**



Universidad Nacional
Autónoma de México

Dirección General de Bibliotecas de la UNAM

Biblioteca Central



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.



FACULTAD DE ESTUDIOS SUPERIORES CUAUTITLÁN
 UNIDAD DE LA ADMINISTRACIÓN ESCOLAR
 DEPARTAMENTO DE EXAMENES PROFESIONALES



UNIVERSIDAD NACIONAL
 AVENIDA DEL
 VALLE

DR. JAIME KELLER TORRES
 DIRECTOR DE LA FES-CUAUTITLÁN
 PRESENTE.

ATN: ING. RAFAEL RODRIGUEZ CEBALLOS
 Jefe del Departamento de Exámenes
 Profesionales de la FES-C.

Con base en el art. 51 del Reglamento de Exámenes Profesionales de la FES-Cuautitlán, nos permitimos comunicar a usted que revisamos el Trabajo de Seminario:

REVIS DE COMPUTADORAS, INSTALACIÓN Y CONFIGURACIÓN DE UN SERVIDOR
DE CORPO ELECTRÓNICO BAJO EL SISTEMA OPERATIVO LINUX.

que presenta el pasante: GONZALE CEJA LEÓN
 con número de cuenta: 9026194-3 para obtener el Título de:
LICENCIADO EN INFORMATICA

Considerando que dicho trabajo reúne los requisitos necesarios para ser discutido en el EXAMEN PROFESIONAL correspondiente, otorgamos nuestro VISTO BUENO.

ATENTAMENTE.
 "POR MI RAZA HABLARA EL ESPIRITU"

Cuautitlán Izcalli, Edo. de México, a 15 de octubre de 19 97

MODULO:	PROFESOR:	FIRMA:
III	ING. ROSA M. MENDOZA	<i>[Firma]</i>
III	M. L. GLORIA PONGT. VILLAGAS	<i>[Firma]</i>
II	ING. MIGUEL ALVAREZ PASAR	<i>[Firma]</i>

DEP/VBOISEN

Agradecimientos.

A mi Madre:

A quien nunca lograré agradecer el amor y apoyo que me ha dado, y sobre todo por sacrificarse tanto para brindarme lo que es para mí una de las cosas más valiosas y difíciles de la vida "Mi Educación".

A mi Padre:

Que a pesar de que hemos estado muy alejados, siempre he contado con su ayuda.

A mis hermanas:

Por todo su cariño, apoyo y por darme la alegría de saber que disfrutan tanto como yo de este logro.

A la memoria de mi sobrino Brian:

Por dejarnos tantos recuerdos lindos y por enseñarme que la vida se vive con mucho entusiasmo, alegría y brindando amor hasta el último momento.

A Anya:

Por ser mi compañera perfecta. Además por enseñarme a no desesperarme y a disfrutar cada una de las desveladas maratónicas que compartimos.

INDICE

INTRODUCCION.	3
OBJETIVOS.	5
1. CONCEPTOS BÁSICOS.	7
1.1 INTRODUCCIÓN A LINUX.	7
1.1.1 HISTORIA DE LINUX.	8
1.1.2 CARACTERÍSTICAS.	10
1.2 SERVICIOS DE INTERNET.	14
1.2.1 SERVICIO DE CORREO ELECTRÓNICO.	15
1.2.2 SERVICIO GOPHER.	15
1.2.3 SERVICIO TELNET.	16
1.2.4 SERVICIO FTP (PROTOCOLO DE TRANSFERENCIA DE ARCHIVOS).	17
1.2.5 WORL WIDE WEB.	18
2. CORREO ELECTRÓNICO.	20
2.1 INTRODUCCIÓN AL CORREO ELECTRÓNICO.	20
2.1.1 CONCEPTO DE CORREO ELECTRÓNICO.	21
2.1.2 SERVIDOR DE CORREO ELECTRÓNICO.	22
2.2 RELACIÓN ENTRE ENLACE DE REDES Y EL CORREO ELECTRÓNICO.	22
2.3 ESTÁNDARES TCP/IP PARA EL SERVICIO DE CORREO ELECTRÓNICO.	24
2.4 DIRECCIONES DE CORREO ELECTRÓNICO.	24
2.4.1 PSEUDO DIRECCIONES DE DOMINIO.	26
2.5 PROTOCOLO SMTP.	26
2.6 EXTENSIÓN MIME PARA DATOS NO ASCII.	28
2.6.1 MENSAJES MIME <i>MULTIPART</i> .	30
2.7. REQUERIMIENTOS DE HARDWARE Y SOFTWARE.	30
3. INSTALACIÓN Y CONFIGURACIÓN.	36
3.1 INTRODUCCIÓN A SENDMAIL.	36
3.2 FUNCIÓN DE SENDMAIL.	36
3.3 EL ARCHIVO SENDMAIL.CF	37
3.4 CONFIGURACIÓN DE SENDMAIL.	56
3.4.1 CONFIGURACIÓN DE SENDMAIL LOCALMENTE.	56
3.4.2 EJECUTANDO SENDMAIL COMO UN <i>DAEMON</i> .	56

3.4.3 COMO ESTABLECER ALIAS.	57
3.4.4 CONFIGURACIÓN DE SENDMAIL PARA MANEJAR CORREO INTERNET.	58
3.4.5 SENDMAIL Y CORREO UUCP.	60
3.4.6 CONFIGURACIÓN DE SENDMAIL PARA MANEJAR CORREO POP	61
3.5 LISTAS DE CORREO.	61
3.6 VERIFICACIÓN DEL FUNCIONAMIENTO DEL CORREO ELECTRÓNICO.	63
4. CASO PRACTICO: INSTALACIÓN Y CONFIGURACIÓN DE UN SERVIDOR DE CORREO ELECTRÓNICO EN LINUX DENTRO DE LA FACULTAD DE ESTUDIOS SUPERIORES CUAUTITLÁN.	66
4.1 INTRODUCCIÓN.	66
4.2 INSTALACIÓN Y CONFIGURACIÓN.	66
4.3 PUESTA EN MARCHA	71
4.4 GESTIÓN DEL SERVIDOR DE CORREO ELECTRÓNICO	73
CONCLUSIONES.	75
BIBLIOGRAFIA	76

INTRODUCCION.

El correo electrónico es una herramienta importante en el ambiente de las comunicaciones, debido a su flexibilidad para enviar y recibir datos, imagen y voz que tal vez lleguen a su destino en cuestión de segundos o minutos, desde redes locales hasta redes mundiales, es decir, se pueden enviar correos electrónicos entre computadoras en red ubicadas en un mismo edificio o enviar mensajes entre equipos que se encuentren en diferentes partes del mundo, siempre y cuando estos formen parte de Internet y cuenten con correo electrónico.

En el presente trabajo se desarrolla toda la metodología y un caso práctico aplicado a la Facultad de Estudios Superiores Cuautitlán, para la instalación y configuración de un servidor de correo electrónico bajo el sistema operativo Linux.

En el primer capítulo se introduce al lector a los conceptos y características básicas del sistema operativo LINUX y de Internet. Esto con el fin de que el lector identifique la interrelación que existe entre estos temas y el correo electrónico, para que se forme una base para la comprensión del presente trabajo.

En el segundo capítulo se tratan aspectos básicos del correo electrónico, como su significado, los principales protocolos que maneja y los requerimientos de hardware-software para su instalación y configuración.

En el tercer capítulo se habla de la metodología para la instalación y configuración del servidor de correo electrónico, como son la instalación de software y la configuración de los archivos principales para que el sistema maneje correo en forma local y global. Además se explica el significado de todos los comandos utilizados para configurar el archivo principal sendmail.cf.

En el último capítulo se aplican todos los conocimientos expuestos anteriormente, para la instalación y configuración de un servidor de correo electrónico dentro de la Facultad de Estudios Superiores Cuautitlán, además se tratan aspectos de la puesta en marcha y la administración del servidor.

OBJETIVOS.

General.

- Describir la metodología que se lleva a cabo en la instalación y configuración del servidor de correo electrónico.

Específico.

- Lograr que la Facultad de Estudios Superiores Cuautitlán cuente con un servicio de correo electrónico óptimo y confiable para los usuarios.

- Identificar las ventajas y desventajas que ofrece el servicio de correo electrónico.

CAPITULO 1

CONCEPTOS BASICOS

1. Conceptos Básicos.

1.1 Introducción a Linux.

Linux es un sistema operativo con muchas características del sistema operativo Unix, es una reimplementación completamente gratuita y a la vez muy potente con especificaciones POSIX (Portable Operating System Interface to Unix), con extensiones de SYSV (Distribuciones de Unix sistema V) y BSD (distribución de Unix de las versiones de Berkeley), lo cual significa que parece Unix pero no proviene del mismo código fuente base, que está disponible tanto en su versión fuente como ya compilada, por esta razón es considerado como un clon de Unix. Las principales diferencias entre estos radican en que Linux está diseñado para procesadores Intel 386 en adelante o compatibles como Cyrix o AMD. Además de que es gratuito y especialmente para PC's. Para el usuario esto resulta una ventaja, ya que puede tener Linux en su casa instalado en su PC aún cuando tenga otros sistemas operativos comerciales como DOS o Windows 95 y aprender tareas de administración de un sistema operativo de red.

Este sistema operativo no es de dominio público, ni es shareware¹. Es software gratuito, usualmente denominado freeware², y se proporcionan copias con el código fuente. Si un usuario redistribuye una modificación está legalmente obligado a proporcionar los fuentes de ésta. Esto debido a la naturaleza del Copyright de GNU (Licencia Pública o GLP), al que Linux está sujeto, sería ilegal no hacerlo así. No obstante es importante saber que es perfectamente legal cobrar por distribuir Linux, mientras se incluya el código fuente.

¹ Software distribuido a través de BBS (Bulletin Board System) y servicios de información libres de cargo sobre una base de prueba. Si se desea utilizar se debe pagar por él.

² Software distribuido sin cargo, se hace uso de él sin pago alguno. La propiedad la retiene el desarrollador que tiene el control de su redistribución.

Por otra parte Linux es una herramienta importante que puede desempeñar tareas de igual o mejor forma que las que realizan sistemas operativos de red, los cuales resultan bastante caros y además por estar respaldados por una marca tienden a encerrar al usuario en el avance de su tecnología y al cambio repentino de software así como de su equipo, lo cual no sucede con Linux, con él no se buscan aspectos lucrativos sino el desarrollo de un sistema operativo de red que pueda dar un buen servicio o apoyo al usuario de éste.

1.1.1 Historia de Linux.

A través del desarrollo de los sistemas operativos más competentes para red, no se había dado el caso de la existencia de un sistema operativo con características tan buenas y muy accesible para el usuario, como es el caso de Linux.

En 1991 comenzó el desarrollo de Linux en Finlandia como proyecto personal del entonces estudiante Linus Torvalds, quien tomó como punto de partida otro viejo conocido, el Minix de Andy. S. Tanenbaum (profesor de sistemas operativos que creó su propio sistema operativo Unix en PCs XT para usarlo en su docencia). Actualmente Linus lo sigue desarrollando, pero a estas alturas el principal autor es la red Internet, desde donde una gigantesca gama de programadores y usuarios aportan diariamente su tiempo aumentando sus prestaciones y dando información y soporte técnico mutuo.

El Copyright de Linux pertenece a Linus B. Torvalds sobre el kernel. Sin embargo, Linux en sí mismo (incluido el kernel) está bajo la GNU Public License (Licencia Pública de GNU o sólo GPL) por lo general referido como Copyleft (libre de derechos) a causa de la diferencia que resulta de los varios acuerdos sobre los derechos de autor.

La GPL requiere que todos los que utilizan elementos que esta cubre para crear programas, hagan disponible el código fuente de esos programas para quienes

pregunten por ellos, e insiste que con el producto se incluya una copia del documento de la GPL. Este acuerdo permite a los programadores ofrecer su software al público en general, al tiempo que conserva los derechos de autor sobre el mismo a fin de continuar con su trabajo. Además, esto asegura que el código se encuentra disponible para otras personas a fin de mejorarlo y modificarlo. La meta es incrementar la suma total de software gratuito disponible.

La versión original comenzó para PCs compatibles (Intel 386 y superiores) con bus ISA, EISA, posteriormente PCI o VLB. MCA (bus de microcanal propietario de IBM) no está actualmente soportado por la falta de documentación.

Actualmente se está portando a distintas plataformas Motorola 680x0 (los Amigas y Ataris), y por ahora funciona bastante bien. Se requiere un 68020 con una MMU (Memory Management Unit), un 68030 o un 68040, y también una FPU (Floating Point Unit). El software de red y X funcionan.

Linux funciona bien en la CPU Alpha de DEC, actualmente bajo "Jensen", "NoName", "Cabriolet", "Universal Desktop Box" (más conocida como Multia), y algunas otras plataformas.

Existe también en desarrollo versiones para otras máquinas, incluyendo MIPS, Sparc, PowerPC y PowerMAC, etc. De todas ellas la más reciente en este momento es la versión para PowerMac (el PowerPC de Apple) basada en el microkernel Mach 3.0 y de la que ya hay una distribución para desarrolladores avalada directamente por Apple y OSF pero conservando el espíritu (gratuito, de libre distribución, etc.) de la versión original.

1.1.2 Características.

•**Multitareas reales:** La multitarea es ejecutar varios procesos a la vez, lo cual no lo hacen sistemas operativos como Windows 95

Estos sistemas operativos permiten realizar varias tareas a la vez pero no reales, a esto se le llama tareas cooperativas, ya que estos procesos no se ejecutan en su totalidad al mismo tiempo, los programas comen hasta que ellos mismos permiten la ejecución de otro programa o no tienen otra cosa que hacer durante ese período. Si trata de iniciar un proceso en segundo plano para dejarlo terminar por si mismo, éste se detiene o avanza lento hasta que se regrese a él.

Por otra parte Linux maneja procesos de multitarea preferente, esto es mientras se da prioridad a la tarea en la que trabaja cada segundo, las que se ejecutan en segundo plano continúan su función hasta finalizar o hasta que necesiten datos de entrada.

•**Multiusuario:** Esto es varios usuarios en la misma máquina al mismo tiempo (¡y sin licencias para todos!).

•**Multiplataforma:** Corre en muchas CPUs distintas, no sólo Intel.

•**Funciona en modo protegido 386.**

•**Cuenta con protección de la memoria entre procesos,** de manera que uno de ellos no pueda bloquear el sistema.

•**Carga de ejecutables por demanda:** Linux sólo lee de disco aquellas partes de un programa que estén siendo usadas actualmente. Muchos sistemas operativos conservan cualquier cosa en la RAM física hasta que se quedan sin espacio y

después comienzan a hacer uso de la memoria virtual (si la poseen). Esto significa que si se deja abierto un proceso y no se hace uso de él durante un tiempo, este ocupa espacio en la RAM física y al cargar nuevos procesos los envía al espacio swap de intercambio. Esto funciona un poco diferente con Linux que maneja la carga por demanda, cuando el proceso se encuentra todavía activo, pero no en uso durante algún tiempo, un sistema Linux lo moverá a sus espacio swap con el fin de liberar más de su RAM física para la atención de nuevos procesos.

•Política de copia en escritura para la compartición de páginas entre ejecutables: esto significa que varios procesos pueden usar la misma zona de memoria para ejecutarse. Cuando alguno intenta escribir en esa memoria, la página (4Kb de memoria) se copia a otro lugar. Esta política de copia en escritura tiene dos beneficios: aumenta la velocidad y reduce el uso de memoria.

•Memoria virtual usando paginación (sin intercambio de procesos completos) a disco: una partición o un archivo en el sistema de archivos, o ambos, con la posibilidad de añadir más áreas de intercambio sobre la marcha (se sigue denominando intercambio, es en realidad un intercambio de páginas). Un total de 16 zonas de intercambio de 128Mb de tamaño máximo pueden ser usadas en un momento dado con un límite teórico de 2Gb para intercambio.

•La memoria se gestiona como un recurso unificado para los programas de usuario y para el caché de disco, de tal forma que toda la memoria libre puede ser usada para caché y éste puede a su vez ser reducido cuando se ejecuten grandes programas.

•Bibliotecas compartidas de carga dinámica (DLL's) y Bibliotecas estáticas.

•Se realizan volcados de estado (core dumps) para posibilitar los análisis post-mortem, permitiendo el uso de depuradores sobre los programas no sólo en ejecución sino también tras abortar éstos por cualquier motivo.

•Mediante un módulo de emulación de iBCS2, casi completamente compatible con SCO, SVR3 y SVR4 a nivel binario.

•Todo el código fuente está disponible, incluyendo el núcleo completo y todos los controladores (drivers), las herramientas de desarrollo y todos los programas de usuario; además todo ello se puede distribuir libremente. Hay algunos programas comerciales que están siendo ofrecidos para Linux actualmente sin código fuente, pero todo lo que ha sido gratuito sigue siendo gratuito.

•Ostenta UNIX POSIX (Portable Operating System Interface) que es un estándar federal de procesamiento de información puesto en marcha por el Instituto Nacional de Estándares y Tecnología de Estados Unidos. Es casi totalmente compatible con POSIX, System V y BSD a nivel fuente. Es decir, que las aplicaciones para otros sabores de Unix se transportan a Linux sin mucha dificultad.

•Pseudo-terminales (pty's).

•Emulación de 387 en el núcleo, de tal forma que los programas no tengan que hacer su propia emulación matemática. Cualquier máquina que ejecute Linux parecerá dotada de coprocesador matemático. Y por supuesto, si un ordenador ya tiene una FPU (unidad de coma flotante), será usada en lugar de la emulación, pudiendo incluso compilar el propio kernel sin la emulación matemática y conseguir un pequeño ahorro de memoria.

•Soporte para muchos teclados nacionales o adaptados y es bastante fácil añadir nuevos dinámicamente.

•Consoles virtuales múltiples: varias sesiones de login a través de la consola entre las que se puede cambiar con las combinaciones adecuadas de teclas (totalmente independiente del hardware de vídeo). Se crean dinámicamente y se puede tener hasta 64.

•Soporte para varios sistemas de archivo comunes, incluyendo minix-1, Xenix y todos los sistemas de archivo típicos de System V, y tiene un avanzado sistema de archivos propio con una capacidad de hasta 4 Tb y nombres de archivos de hasta 255 caracteres de longitud.

•Acceso transparente a particiones MS-DOS (o a particiones OS/2 FAT) mediante un sistema de archivos especial: no se requiere un comando especial para usar la partición MS-DOS, parece un sistema de archivos normal de Unix (excepto por algunas restricciones en los nombres de archivo, permisos, etc.). Las particiones comprimidas de MS-DOS 6 no son accesibles en este momento, y no se espera que lo sean en el futuro. El soporte para VFAT (WNT, Windows 95) ha sido añadido al núcleo de desarrollo y estará en la próxima versión estable.

•Un sistema de archivos especial llamado UMSDOS que permite que Linux sea instalado en un sistema de archivos DOS.

•Soporte en sólo lectura de HPFS-2 del OS/2 2.1

•Sistema de archivos de CD-ROM que lee todos los formatos estándar de CD-ROM.

•TCP/IP, incluyendo ftp, telnet, e-mail, NFS, etc. Muchos sistemas operativos de PC no tienen incorporado el protocolo TCP/IP para la comunicación a Internet. Para la conexión y conversión de los propios protocolos de comunicación requieren de

programas auxiliares. Linux utiliza TCP/IP para su propia conectividad en red; por consiguiente no se requiere un programa de conversión auxiliar. Esto hace más rápido el acceso a Internet y existe menos oportunidad de error porque sus procesos no tienen que atravesar muchas capas de programas.

- Appletalk disponible en el actual núcleo de desarrollo.

- Software cliente y servidor Netware disponible en los núcleos de desarrollo.

1.2 Servicios de Internet.

Internet es una red mundial, compuesta por un conjunto de equipos conectados en redes de diferentes tipos, utilizando protocolos como TCP/IP, que identifican los datos aunque procedan de diferentes equipos (PC's, Macintosh, Amiga) y usen sistemas operativos incompatibles.

Con el crecimiento de Internet y el desarrollo de tecnología obligó a los desarrolladores de computadoras, sistemas operativos y programas de aplicación, a crear servicios especiales que les permitieran encontrar y transferir la información sin grandes esfuerzos.

Algunos de estos programas, desde entonces diseñados bajo el esquema cliente/servidor, utilizando el sistema operativo UNIX fueron Mail o correo electrónico, Telnet, Gopher, FTP y finalmente World Wide Web. Muchos de estos servicios se ejecutan por medio de un programa *daemon* (demonio), es decir, existe un programa que se ejecuta en segundo plano y está pendiente de los elementos que administra. Un ejemplo de *daemon* es el programa Sendmail de correo electrónico.

1.2.1 Servicio de Correo Electrónico.

El correo electrónico difiere de otras aplicaciones porque no es un servicio de usuario a usuario: no es necesario que las máquinas emisora y receptora del correo electrónico se comuniquen directamente entre sí. Al correo electrónico se le conoce como un servicio de "almacenaje y reenvío". El correo pasa de una máquina a otra hasta que llega a su destino final.

El correo es un servicio de los más útiles. Todos los usuarios de Internet están identificados con una dirección de correo electrónico, que se construye normalmente mediante el identificador del usuario en el ordenador que le da acceso a la red, seguido del símbolo '@' (arroba) y del nombre del dominio de la máquina. Por ejemplo, si nuestro dominio es servidor.unam.mx, las direcciones de los usuarios tendrán el siguiente formato: usuario@servidor.unam.mx (siempre en minúsculas).

Para enviar correo sólo es necesario escribir el mensaje e indicar la dirección del destinatario. La transferencia del correo no se realiza mediante una conexión directa con el ordenador del receptor por lo que podemos enviar correo sin tener que preocuparnos de si el destinatario está conectado a la red, si no lo está la próxima vez que se conecte recibirá el correo pendiente, del mismo modo nosotros recibiremos nuestro correo cuando nos conectemos a nuestro respectivo proveedor.

1.2.2 Servicio Gopher.

Este programa cliente/servidor fue desarrollado en Minesota en 1991. Gopher está basado en una serie de menús que van ofreciendo una serie de archivos, los cuales se pueden bajar a la computadora a través del protocolo Gopher, que es otro más del paquete de protocolos de TCP/IP.

Gopher fue uno de los primeros servicios de Internet que realizó un intento para ofrecer una interfaz amigable al usuario. Es un servicio que permite tener acceso a información al hacer selecciones de una serie de menús. Cuando nos conectamos con un sitio que proporciona servicio al Gopher da un menú de alternativas disponibles. Luego se puede seleccionar la alternativa del menú sin conocer el nombre o la dirección IP del sitio de destino o del directorio y los nombres de archivos de la información. Gopher maneja este tipo de aspectos con transparencia.

Una desventaja de Gopher es que no hay una lista de temas estándares para los diversos servidores Gopher. Los administradores de cada servidor Gopher individual han organizado su información a su manera. Es decir, que cada servidor Gopher a que se tiene acceso incluye diferentes temas. Si el servidor no presenta algunos de los mismos temas es probable que los haya nombrado de forma diferente.

No hay ningún recurso de información en Internet que sea en la actualidad específico de Gopher. Cualquier cosa que se logre por medio de Gopher también puede obtenerse por otros medios, como FTP o Telnet. En algunos casos, los sitios quizá escojan hacer disponibles los recursos sólo por medio de Gopher, por razones de seguridad.

1.2.3 Servicio Telnet

Telnet es un programa que permite ingresar a una computadora conectada a la red, aunque se encuentre ubicada en un sitio lejano. Una vez aceptada su estancia o conexión en el sistema, se puede hacer uso de los recursos y ejecutar comandos, como si se estuviera registrado localmente, excepto a los que sean de acceso restringido. La mayoría de los servicios Telnet se proporcionan mediante menús, lo que hace más sencilla su utilización.

En esta aplicación se permite la conexión remota con cualquier ordenador de la red, como si nuestro ordenador fuera un terminal de esa máquina, lo que nosotros

tecleamos se transmite al ordenador remoto que lo procesa y nos devuelve el resultado.

Para conectarse a una computadora por medio de Telnet, se debe conocer un nombre de usuario y una contraseña validos de la máquina remota. Aunque algunos sistemas proporcionen capacidades de registro para huéspedes, casi siempre ponen al usuario en un *shell* restringido o en un sistema de menú. La idea de esto es proporcionar seguridad a la computadora y proteger el sistema contra usuarios desconocidos y mal intencionados. Un *shell* restringido impide que un usuario ejecute comandos específicos, mientras que un sistema de menú permite sólo alternativas de un juego de menús predefinido bloqueando por completo el acceso al *shell*.

1.2.4 Servicio FTP (Protocolo de Transferencia de archivos).

FTP (File Transfer Protocol) son las siglas del protocolo estándar para la transferencia de ficheros. Mediante un programa FTP o a través de WWW podemos acceder a los ficheros de todas las máquinas conectadas a Internet y que actúan como servidores de ficheros. El proceso normal para comunicarnos con un servidor FTP es indicarle a nuestro programa la dirección del ordenador y en unos instantes nos preguntará el usuario y la clave, prácticamente todos los servidores FTP soportan usuarios anónimos, es decir, no nos hace falta tener una clave para acceder a ellos simplemente cuando nos pregunta el nombre de usuario le respondemos *anonymous* y cuando nos solicita la clave le damos nuestra dirección de correo electrónico (si accedemos desde WWW este proceso se realiza de forma automática).

FTP tiene una especial importancia en la super-red, ya que es el protocolo que más se utiliza en la transferencia de archivos entre computadoras conectadas a la red de Internet. Este protocolo permite intercambiar archivos entre computadoras de

las más diversas arquitecturas, que utilizan una gran cantidad de sistemas operativos diferentes.

Como la mayoría de los servicios de Internet que se soportan sobre la plataforma UNIX, FTP es una serie de programas cliente/servidor que permiten acceder a una red o nodo y pedirle que permita transferir uno o varios archivos a la computadora, mediante una serie de reglas o normas de control de comunicación que se ha denominado FTP, y que forman parte del conjunto global de protocolos de TCP/IP.

El principio básico de FTP es que cuando se desea bajar un archivo a la computadora, el procedimiento se denomina bajar (Download) archivos, y la operación inversa se conoce como subir (Upload) archivos. Ese mismo principio utiliza el servicio Telnet para comunicarse de una computadora a otra y transferir información entre ellas.

FTP es una de las mejores herramientas de Internet que ha evolucionado en las interfaces gráficas. Siempre se necesitará transferir un archivo a la computadora y será FTP el medio más adecuado para hacerlo.

1.2.5 World Wide Web.

World Wide Web (WWW) es la herramienta de navegación más reciente, esta basada en documentos en formato hipertexto y pueden contener imágenes, sonido, animaciones, etc. Se puede viajar por la red de un documento a otro, dispersos por todo el mundo, mediante referencias cruzadas. Esto quiere decir que la selección de un término en una página nos puede llevar a otro documento situado en otra máquina a miles de kilómetros.

Además de todo esto a través de WWW tenemos acceso a servidores de información Gopher, FTP o a establecer conexiones Telnet cuando es necesario.

CAPITULO 2
CORREO ELECTRONICO

2. Correo Electrónico.

2.1 Introducción al Correo Electrónico.

El correo electrónico es uno de los servicios de aplicación disponibles más ampliamente utilizados como en la mayor parte de los servicios TCP/IP, utiliza el modelo cliente/servidor. Los buffers del sistema de correo de los mensajes que entran y salen permiten que la transferencia desde el cliente y el servidor se realice como un proceso subordinado.

E-mail es popular porque ofrece un método rápido y flexible de transferencia de información. Su flexibilidad se refleja en su adaptación para enviar pequeñas notas y grandes y voluminosos documentos mediante un mecanismo sencillo. Muchos usuarios ocupan este medio para enviar archivos en vez de utilizar los protocolos de transferencia de archivos.

Los protocolos de red envían paquetes directamente a sus destinos, utilizando límite de tiempo y retransmisión para los segmentos individuales si no se devuelve un acuse de recibo. Sin embargo, en el correo electrónico, el sistema debe proporcionar los medios cuando la máquina remota o las conexiones de la red han fallado. El emisor no desea esperar a que la máquina remota este disponible para continuar trabajando, ni el usuario quiere que se aborte la transmisión sólo porque las comunicaciones con la máquina remota no están disponibles temporalmente.

Para manejar las entregas con retraso, el sistema de *e-mail* utiliza una técnica conocida como *spooling*. Cuando el usuario envía un mensaje de correo, el sistema coloca una copia en su área de almacenamiento privado (*spool*) junto con la identificación del emisor, recipiente, máquina de destino y hora de depósito. El

sistema, entonces inicia la transferencia hacia la máquina remota como una actividad subordinada o secundaria, permitiendo al emisor que continúe con otras actividades.

El conjunto de protocolos TCP/IP proporciona estándares separados para el formato de los mensajes de correo y la transferencia de correo. El formato de mensaje de correo conocido como 822 utiliza una línea en blanco para separar el encabezado del mensaje del cuerpo del mensaje. El Simple Mail Transfer Protocol (SMTP) define como un sistema de correo en una máquina transfiere correo hacia el servidor en otra.

Las Multipurpose Internet Mail Extensión (MIME) proporcionan un mecanismo que permite que datos arbitrarios se transfieran mediante el SMTP. MIME añade líneas al encabezado de un mensaje *e-mail* para definir el tipo de datos y la codificación utilizada. El tipo multipart mezclado de MIME permite que un sólo mensaje contenga varios tipos de datos.

2.1.1 Concepto de Correo Electrónico.

El correo electrónico es cualquier programa que emplean los usuarios en un sistema de computadora aislada o en un sistema de red, para enviar y recibir mensajes. Al programa se le proporciona, por lo menos, la dirección del destinatario y el mensaje que se quiere enviar. La dirección tiene el nombre de registro de la persona que recibirá el correo. Si ese usuario se encuentra en otro sistema de una red, la dirección también incluye un medio para identificar el sistema de computadora de destino. El mensaje se prepara mientras se utiliza el programa de correo electrónico o antes mediante un editor de texto.

2.1.2 Servidor de Correo Electrónico.

El servidor de correo electrónico es un equipo dedicado a dar diversos servicios de correo electrónico como el intercambio de mensajes, el manejo de listas y también es quien almacena todos los mensajes para después distribuirlos a sus debidos destinatarios.

2.2 Relación entre enlace de redes y el correo electrónico.

Existen dos diferencias principales entre sistemas de computadoras comerciales para enviar correo electrónico que no están conectados a Internet y los sistemas conectados a la superred por medio de TCP/IP.

En primer lugar, una red TCP/IP hace posible el servicio de entrega universal, puesto que proporciona interconexión global entre una máquina y otra, es decir, toda máquina que se encuentra conectada a una red de redes se comportan como si estuvieran conectadas a una sola red independiente de los equipos específicos de ciertos vendedores.

En segundo lugar, el sistema de correo electrónico construido en el TCP/IP es inherentemente más confiable que los construidos a partir de redes arbitrarias, esto es, el uso de TCP/IP hace la entrega de correspondencia más confiable que otros mecanismos. El TCP/IP proporciona conectividad de extremo a extremo, lo que significa que el software de correo en la máquina emisora actúa como un cliente, contactando a un servidor en el destino final. Sólo después que el cliente logra transferir un mensaje de correo al servidor, se elimina el mensaje de la máquina local. De esta manera se refuerza el siguiente principio:

*"Los sistemas de correo que utilizan la entrega de extremo a extremo pueden garantizar que cada mensaje de correo se mantenga en la máquina del emisor hasta ser copiado con éxito a la máquina del receptor"*³.

Para la transferencia de mensajes de correo electrónico a menudo se emplean compuertas de correo electrónico (*Mail gateways*). En cada sistema, la máquina del emisor no establece contacto directamente con la máquina del receptor, sino que envían el correo a través de una o más máquinas intermedias que completan el envío.

La principal desventaja de utilizar compuertas de correo es que reducen la confiabilidad. Una vez que la máquina del emisor transfiere un mensaje a la primera máquina intermedia, se descarta la copia local. Así, mientras el mensaje está en tránsito, ni el receptor ni el emisor tienen una copia. Si se dan fallas entre las máquinas intermedias esto puede provocar la pérdida del mensaje sin que se informe ni al emisor ni al receptor. También se puede dar la pérdida de un mensaje cuando una de las compuertas de correo realiza un mal ruteo del mensaje. Otra desventaja de las compuertas de correo es que manejan un retardo. Es decir, manejan mensajes por minutos, horas e incluso días, si no puede enviar el mensaje a la siguiente máquina. Ni el emisor ni el receptor pueden determinar en qué lugar se ha retrasado el mensaje, por qué no ha llegado o con qué retardo llegará. Bueno, el punto que hay que resaltar es que tanto el emisor como el receptor dependen de máquinas intermedias de las cuales no tienen ningún control.

Se puede pensar que si las compuertas reducen la confiabilidad por qué se usan entonces, bien, el principal motivo por lo que se utilizan estas compuertas de correo es la interoperabilidad. Debido a que proporcionan conexiones entre el estándar

³E. Comer Douglas. *Redes Globales de Información con Internet y TCP/IP*. 3ª ed. México: Prentice Hall, 1996. P 443.

TCP, el sistema de correo TCP/IP estándar y otros sistemas de correo, así como redes que no soportan protocolos de Internet.

2.3 Estándares TCP/IP para el servicio de correo electrónico.

El objetivo principal del protocolo TCP/IP es reforzarse por proporcionar interoperatividad a través de un amplio rango de sistemas de computadoras y redes. Para extender la interoperatividad del correo electrónico, el TCP/IP divide sus estándares de correo en dos grupos. Un estándar especifica el formato del encabezado para los mensajes de correo. El otro especifica los detalles del intercambio de correo electrónico entre dos computadoras. Mantener los dos estándares separados para el correo electrónico hace posible construir compuertas de correo que conectan redes de redes TCP/IP con algunos sistemas de entrega de correo de otros vendedores, siempre y cuando utilicen el mismo formato de mensajes para ambos.

El formato de los mensajes de correo ha sido seleccionado para facilitar el proceso y realizar el transporte a través de máquinas heterogéneas. Mantener el formato del encabezado de correo sin cambios permite utilizarlo dentro de un amplio rango de sistemas. La restricción de los mensajes al formato de sólo texto evita los problemas de seleccionar una representación binaria estándar y traducir entre la representación estándar y la representación de la máquina local.

2.4 Direcciones de Correo Electrónico.

Los formatos de las direcciones de correo electrónico varían entre sistemas de e-mail. Así, puede ser difícil determinar una dirección de correo electrónico correcta. Dentro de la red global de Internet, la dirección de correo tienen una forma simple y fácil de recordar por ejemplo:

local-part@domain-name.

Donde domain-name es el nombre del dominio de un destino de correo al que el correo debe ser entregado y local-part es la dirección de un buzón en la máquina.

Por otra parte, las computas de correo electrónico vuelven las direcciones más complejas. Cualquiera que esté fuera de Internet debe direccionar el correo hacia la compueta de correo más cercana o tener software que lo haga de manera automática. Por ejemplo si se opera una compueta de correo que conecta redes exteriores con Internet, algunas direcciones son de la siguiente forma:

comer%@purdue.edu@relay.cs.net

En esta dirección una compueta actúa primero alcanzando o conectando la máquina relay.cs.net, el software de compueta de correo extraerá después la local-part, cambiando el signo de (%) por el signo (@) y de esta forma se obtiene una nueva dirección destino para enviar la correspondencia.

La razón por la que las direcciones se hacen complejas cuando se incluyen localidades que no son de Internet, es que la función de transformación de las direcciones de correo electrónico por medio de las compuetas se hace localmente para cada máquina.

Debido a que cada compueta de correo electrónico determina los detalles exactos de cómo interpretar y transformar las direcciones de correo, no hay un estándar para los direccionamientos que cruzan las compuetas de correo hacia redes que están fuera de Internet.

2.4.1 Pseudo direcciones de dominio.

Para ayudar a resolver el problema de los diversos sistemas de correo, cada uno con su propio formato de dirección, una localidad puede utilizar los nombres de dominio-tipo para todos los direccionamientos de e-mail, aun cuando la localidad no utilice el sistema de nombres de dominio. Es decir, una localidad que emplee el protocolo UUCP (COPY UNIX TO UNIX) puede implementar un pseudo-dominio uucp, que permita especificar a los usuarios direcciones de correo de la forma:

dirección tipo-uucp@uucp o en la forma relacionada: usuario@localidad-uucp.uucp.

El software para el envío de correo local reconoce las direcciones especiales y las transforma en sintaxis de dirección requerida por el software de red UUCP. Esto resulta una ventaja para el usuario porque todas las direcciones electrónicas tienen el mismo formato general, independientemente de la red de comunicaciones subyacente utilizada para llegar hasta el recipiente. Aunque el pseudodominio de las direcciones de correo tienen la misma forma que los nombres de dominio, sólo pueden utilizarse con el correo electrónico no se pueden encontrar direcciones IP o direcciones de intercambio de correo para éstas, utilizando el sistema de nombres de dominio.

2.5 Protocolo SMTP.

El protocolo TCP/IP especifica un estándar para el intercambio de correo entre máquinas. Es decir, el estándar especifica el formato exacto de los mensajes a un cliente en una máquina que lo utiliza para transferir correo hacia el servidor en otra.

Existe otro protocolo de transferencia estándar que forma parte de TCP/IP, este se conoce como SMTP, Simple Mail Transfer Protocol (Protocolo de transferencia de correo simple). El protocolo SMTP se enfoca específicamente en cómo transfiere el

sistema de entrega de correo subyacente los mensajes a través de un enlace de una máquina a otra. No especifica de qué manera acepta el sistema de correo los mensajes de correo de un usuario o cómo presenta al usuario la interfaz de usuario el correo entrante. El SMTP tampoco especifica en qué forma se almacena el correo o con qué frecuencia el sistema de correo trata de enviar mensajes.

Como se puede ver este protocolo es sumamente sencillo. La comunicación entre un cliente y un servidor consiste en texto ASCII que es posible leer. Inicialmente, el cliente establece una conexión de flujo confiable con el servidor y espera que el servidor le envíe un mensaje parecido a este (220 READY FOR MAIL) para saber que se encuentra listo para atender su llamado. (Si el servidor está sobrecargado deberá retardar el envío del mensaje 220 temporalmente). Al recibir el mensaje 220, el cliente envía un comando HELO. El extremo de una línea marca el fin de un comando. El servidor responde identificándose. Una vez que la comunicación se ha establecido, el emisor puede transmitir mensajes de correo, terminar la conexión o solicitar al servidor que intercambie las funciones emisor y receptor para que los mensajes puedan fluir en la dirección opuesta. El receptor debe enviar un acuse de recibido por cada mensaje. También puede abortar la conexión completa o la transferencia del mensaje actual.

Las transacciones de correo comienzan con un comando *MAIL* que proporciona la identificación del emisor así como un campo *FROM:* que contiene la dirección en la que los errores se deberán reportar. Un recipiente prepara su estructura de datos para recibir un nuevo mensaje de correo y responde al comando *MAIL* enviando la respuesta 250. La respuesta 250 significa que todo está bien. Luego de un comando *MAIL* exitoso, el emisor emite una serie de comandos *RCPT* que identifican a los recipientes del mensaje de correo. Los receptores deben enviar un acuse de recibido por cada comando *RCPT* enviando un 250 OK o el mensaje de error 550 No such user here.

Después de que todos los comandos RCPT han sido reconocidos, el emisor emite un comando DATA. Este comando informa al receptor que el emisor está listo para transferir un mensaje de correo completo. El receptor responde con el mensaje 354 Start Mail Input y especifica la secuencia de caracteres utilizada para determinar el mensaje de correo. El fin de la secuencia consiste en cinco caracteres: retorno de carro, alimentación de línea, punto, retorno de carro y alimentación de línea.

2.6 Extensión MIME para datos no ASCII.

Un punto importante en el correo electrónico, es poder leer y recibir por mensajería electrónica textos escritos en español sin deformar su contenido ni perder los acentos y símbolos propios.

El problema se reduce a lo de siempre: el uso de caracteres especiales no incluidos en la tipografía básica sajona requiere trabajar con 8 bits, pero aún hay muchos sistemas que siguen trabajando con 7 bits, lo que trunca o deforma mensajes escritos sin las debidas precauciones.

En el estándar RFC822 que define el formato usado en los mensajes de correo de Internet no se permite el uso de 8 bits en las cabeceras de los mensajes, dejando abierta la posibilidad de 8 bits en el cuerpo del mensaje. Pero tampoco nos podemos encerrar en esta posibilidad ya que no tenemos garantías de que lo respete cualquier sistema que pueda recibir nuestros documentos. Una solución cada vez más extendida, y que permite a los usuarios escribir con libertad y enviar prácticamente cualquier información dejando todos esos detalles al software de gestión del correo (tanto los agentes de usuario como los agentes de transporte en los servidores de correo) es mediante el estándar MIME, que permite codificar nuestros mensajes de 8 bits y reducirlos a caracteres de 7 bits que puedan pasar a través de cualquier sistema sin peligro de perder información. El inconveniente (cada vez menor a medida que este sistema se extienda más y más) es que si

intentamos leer un mensaje con codificación MIME sin un software que no implemente este sistema y no pueda interpretarlo, puede que veamos "basura" intercalada con nuestro mensaje, o que incluso no entendamos absolutamente nada.

MIME deja toda la responsabilidad al software, por lo que si ambos extremos (remite y destinatario) usan software compatible con MIME, el usuario puede enviar junto a textos simples cualquier combinación de tipos de ficheros (documentos generados por procesadores de textos, gráficos, sonido, vídeo, etc.). Ni siquiera es necesario que ambos usen el mismo software. Basta con que ambos sean capaces de generar la codificación durante el envío, y de interpretarla en la recepción.

Para permitir la transmisión de datos no ASCII a través de *e-mail*, la IETF (Fuerza de Tarea de Ingeniería Internet) definió la Multipurpose Internet Mail Extension (MIME). La MIME no cambia al SMTP ni lo reemplaza. De hecho, la MIME permite que datos arbitrarios sigan codificándose en ASCII y luego se envían por medio de mensajes *e-mail* estándar.

La información de MIME reside en el encabezado de correo 822 la línea del encabezado MIME que especifica la versión de MIME utilizada, el tipo de datos que se envían y la codificación empleada para convertir los datos en ASCII.

El estándar MIME especifica que una declaración *Content-Type* debe contener dos identificadores: un tipo de contenido y un subtipo, separados por una diagonal. El estándar define siete tipos de contenidos básicos, los subtipos válidos para cada uno y las codificaciones de transferencia. Además de los tipos estándar y los subtipos, permite a un emisor y a un receptor definir tipos de contenido privado.

Los tipos básicos son los siguientes: text, image, audio, vídeo, application, multipart y message.

2.6.1 Mensajes MIME *multipart*.

El tipo de contenido *multipart* de MIME es útil pues añade una flexibilidad considerable. El estándar define cuatro posibles subtipos para un mensaje *multipart*, cada uno proporciona una funcionalidad importante. El subtipo *mixed* permite que un solo mensaje contenga submensajes independientes, de los que cada uno tiene un tipo independiente y una codificación diferente. Los mensajes *multipart* mezclados hacen posible incluir textos, gráficos y audio en un solo mensaje, o permiten el envío de un memorándum con segmentos de dato adicionales asociados. El subtipo *alternative* permite que un solo mensaje incluya varias representaciones de los mismos datos. Algunas alternativas de los mensajes *multipart* son útiles cuando se envía un memorándum a muchos recipientes de los que no todos utilizan el mismo hardware y software de sistema. El subtipo *parallel* permite que un solo mensaje incluya subpartes que deben ser vistas juntas (por ejemplo, subpartes de audio y vídeo que deben presentarse de manera simultánea). Por último, el subtipo *digest* permite que un solo mensaje contenga un conjunto de otros mensajes (por ejemplo, la colección de mensajes e-mail de una discusión).

2.7. Requerimientos de Hardware y Software.

Los requerimientos hardware con respecto al servidor de correo electrónico, van a depender por el número de servicios a ofrecer y al número de usuarios a atender.

Linux soporta el siguiente Hardware:

CPU:

Cualquiera que pueda ejecutar programas en modo protegido del 386 (todos los modelos de 386, 486, 586 y 686 deberían valer; los 286 no funcionan ni lo harán jamás). Se está portando a PowerPC, ARM y a arquitecturas MIPS.

Arquitecturas:

Buses ISA o EISA. MCA (el de los PS/2) no funciona. Los buses locales (VLB y PCI) funcionan.

RAM:

Teóricamente hasta 1 GB, aunque jamás ha sido probado hasta este extremo. Algunas personas (incluido Linus) han notado que añadiendo memoria sin incrementar el caché al mismo tiempo su máquina se ha vuelto muy lenta extremadamente, por ello si se añade memoria y se encuentra la máquina más lenta hay que probar agregándole más caché. Por encima de 64MB de RAM se requiere un parámetro de arranque, ya que la BIOS no puede detectar más que 64 MB por su implementación.

Almacenamiento de datos:

Las unidades tipo AT (IDE, controladores de disco duro de 16 bits con MFM o RLL, o ESDI) son soportadas, como también lo son los discos duros y CD-ROM SCSI, con un adaptador SCSI soportado. Las controladoras tipo XT (controladoras de 8 bits con MFM o RLL) también son soportadas. Adaptadores SCSI soportados: Adaptec 1542, 1522, 1740 y la serie 27xx, controladoras Buslogic vía compatibilidad con Adaptec o con su propio driver, controladoras basadas en NCR53c810, Seagate ST-01 y ST-02, de Future Domain la serie TMC-88x (o cualquier placa basada en el chip TMC950) y TMC1660/1680, Ultrastor 14F, 24F y 34F, Western Digital wd7000 y otras. Las unidades de cinta SCSI y algunas QIC-02 y QIC-80 también son soportadas. Bastantes unidades de CD-ROM tienen soporte también, incluyendo Matsushita/Panasonic, Mitsumi, Sony, Soundblaster, Toshiba, ATAPI y otras.

Video:

VGA, EGA, CGA y Hercules (y compatibles) trabajan en modo texto. Para gráficos y X hay soporte para (al menos) VGA normal, algunas tarjetas super-VGA (la mayoría de las basadas en ET3000, ET4000, Paradise, y algunas Trident), S3, 8514/A, ATI MACH8, ATI MACH32 y Hercules. (Linux usa XFree86 como servidor X y éste es quien determina qué tarjetas son soportadas).

Redes:

Western Digital 80x3, ne1000, ne2000, 3com503, 3com509, 3com589 PCMCIA, Allied Telesis AT1500, la mayoría de las placas LANCE, adaptadores de bolsillo d-link, PPP, SLIP, CSLIP, PLIP (Parallel Link IP) y más.

Otro hardware:

SoundBlaster, ProAudio Spectrum 16, Gravis Ultrasound, varios tipos de ratón de bus (Microsoft, Logitech, PS/2).

Bien, el hardware listado anteriormente fue con respecto a compatibilidad y es opcional, ahora trataremos aspectos de hardware enfocados a el número de servicios y usuarios.

Los requerimientos de la CPU en relación con el número de usuarios se muestran en la tabla 2.1.

Tabla 2.1 Requerimientos de CPU.

CPU	USO
486 DX 33Mhz	Dos usuarios y lista de correo.
486 DX-2 66Mhz	Cinco usuarios. Lista de correos pequeña. Compilación
Pentium	Diez o más usuarios. Tres listas de correo. Dos compilaciones.

Requerimientos de RAM en relación con el número de usuarios y requerimientos de uso se muestran en la tabla 2.2.

Tabla 2.2 Requerimientos de RAM

USUARIOS	USO	CANTIDAD RAM
Dos	Lista de correo. Entrada de noticias	8 Mb
Cinco	Dos listas de correo. Entrada de noticias. Solicitudes de páginas Web	16 Mb
Diez	Tres listas de correo. Entrada de noticias. Solicitudes de páginas Web y solicitudes Gopher.	24 Mb

Elección de Disco Duro.

La mayoría de las unidades de disco duro que las personas adquieren son IDE. Estas unidades son buenas para muchos usuarios de computadoras, pero éstos sólo tienen acceso a un archivo a la vez. Si se va a ejecutar un dominio grande se debe considerar un disco SCSI. Las unidades de disco duro SCSI pueden tener acceso a múltiples archivos a la vez; da mayor rapidez y eficiencia en el acceso a los archivos. Acelera los procesos de correo y de noticias de forma increíble.

La capacidad del disco requerido es de 1GB, ya que es el mínimo que se está manejando en el mercado actualmente y es suficiente espacio para realizar un buen servicio de correo.

Con respecto al software, sólo se requiere la instalación de cualquier distribución de Linux para PC, como puede ser Slackware, Debian o RedHat en cualquiera de sus versiones; dando preferencia a la instalación completa de las series referentes a red, para que de esta forma se carguen los archivos de configuración para el correo como son el Sendmail y software para correo UUCP y POP.

CAPITULO 3
INSTALACION Y
CONFIGURACION

3. Instalación y Configuración.

3.1 Introducción a Sendmail.

El Sendmail es uno de los programas de correo electrónico más usados actualmente. El Unix SCO incluye por default el programa Sendmail, este es muy usual para la configuración de correo local, Internet y UUCP. Hay algunas variaciones en el Sendmail básico como el Sendmail+IDA, que es muy usado.

La variedad de programas y protocolos usados por el e-mail, complican su configuración y el soporte. El Sendmail es un programa muy poderoso de correo, pero también tiene muchas dificultades para aprender y entender su configuración. Por otra parte Sendmail+IDA es diferente. Este remueve la necesidad de editar siempre los secretos del archivo sendmail.cf y permite al administrador definir el sitio específico de enrutamiento y la configuración de direcciones de una forma fácil por medio de archivos llamados tables. Con esto el Sendmail+ida permite ahorrar horas de trabajo y stress.

3.2 Función de Sendmail.

El Sendmail acepta correo desde programas de correo de otros usuarios, interpreta de direcciones de correo, reescribe las direcciones de una forma apropiada para los programas de distribución, y las rutas de correo para la correcta distribución. Del mismo modo para la entrada de correo Sendmail interpreta las direcciones y distribuye el correo a los usuarios o envía los mensajes a otros sistemas.

Entre otras funciones de Sendmail esta:

El envío y distribución de correo SMTP (Internet).

Proporciona un amplio sistema de alias de correo, manejar noticias (news), así como listas de correo.

3.3 El archivo *sendmail.cf*

El archivo se divide en secciones específicas, y cada sección está etiquetada para indicar su contenido. (Estas etiquetas están indicadas por el signo de (#)), se listarán a continuación algunos consejos para la comprensión de la estructura del archivo *sendmail.cf*.

•Cualquier elemento que sea específico para la máquina en la que trabaje se encuentra con probabilidad al principio del archivo.

•Los comandos están agrupados de acuerdo con el tipo a que pertenezcan, por tanto, los comandos que son parecidos estarán en la misma área del archivo.

•La mayor parte de un archivo *sendmail.cf* esta constituido por reglas para reescribir, las cuales se emplean para volver a escribir direcciones de correo electrónico cuando es necesario.

•La última parte del archivo está integrada por definiciones de remitente y sus reglas para reescribir asociadas. Las definiciones de remitente indican a Sendmail a donde podría dirigirse el correo y si se trata de un usuario o un programa.

•El primer carácter de una línea es el comando para esa línea, ya que las variables y comandos Sendmail constan sólo de un carácter.

El archivo `sendmail.cf` se divide en secciones y tiene la siguiente estructura:

Información local. Se define la información que está especificada en el host individual. Esta sección es usualmente configurada típicamente en la instalación de Linux.

Macros Generales. Define la información que es especificada para la red local. Esta sección es modificada usualmente durante la configuración.

Clases. Define grupos de nombres de host o nombres de dominio usados para rutinas especiales de correo. No se requiere de su modificación normalmente.

Número de versión. Identifica el número de versión del archivo `sendmail.cf`. Incrementa el número de versión cada vez que se modifica la configuración.

Macros especiales. Define algunas macros especiales usadas por Sendmail. También se especifica información acerca de los protocolos utilizados para el intercambio de correo. Esta sección no es modificada.

Opciones. Define las opciones Sendmail. Esta sección normalmente no requiere modificaciones.

Mensajes de prioridad. Define varios mensajes con valor prioritario para Sendmail. Esta sección no es modificada.

Usuarios confiables. Define quienes son los usuarios confiables que sobrepasan direcciones remitentes cuando ellos envían correo. Esta sección no es modificada. Añadir usuarios a esta lista resulta un problema de seguridad.

Formato de encabezado. Define el formato de encabezado que Sendmail inserta en el mensaje de correo. Esta sección no es modificada.

Reglas para reescribir. Define las reglas usadas para reescribir direcciones de correo. Esta sección no es modificada durante el inicio de la configuración de Sendmail. Únicamente se modifica para corregir un problema o añadir un nuevo servicio.

Remitente. Define las instrucciones usadas por Sendmail para invocar los programas de distribución de remitentes. Esta sección normalmente no se modifica.

Regla nula establecida. Define una reescripción de regla especial llamada regla nula establecida, que es aplicada para la distribución de direcciones. Esta sección no es modificada. La parte de la regla nula establecida que es modificada tiene su propia sección de encabezado.

Parte de Máquina-dependiente de la regla nula establecida. Define las partes de la regla nula establecida que son especificadas para la configuración. Esta sección varía ligeramente sobre la configuración basada del sistema. Un sistema que puede distribuir directamente correo UUCP y SMTP tendrá diferentes reglas de descripción mas que un sistema que pueda solamente distribuir correo UUCP.

La sintaxis de los comandos de Sendmail es difícil descifrar, pero algo que hay que recordar es que los comandos sólo constan de un carácter en su archivo `sendmail.cf` y las primeras letras de las líneas son el comando. Estos comandos son los siguientes:

- "R y S definen reglas para reescribir"

- "The left-hand side (LHS)" (lado izquierdo)

• "The right-hand side (RHS)" (lado derecho)

• "D define macros"

• "C y F definen macros de clase"

• "M define remitente"

• "H define encabezados"

• "O define opciones"

• "P define prioridades"

• "T declara usuarios confiables"

R y S reglas para reescribir.

Son el núcleo del archivo `sendmail.cf`. Estas tienen una lista ordenada de formatos de reposición, las cuales son aplicadas a cada dirección. El comando `Sendmail` busca a través de las reglas establecidas para una igualdad sobre el lado izquierdo (lhs) de la regla. Cuando una regla es igual, la dirección es reemplazada por el lado derecho (rhs) de la regla.

Hay varias reglas de reescripción establecidas. Algunas son usadas internamente y deben de tener una semántica específica. Otras no deben de tener una semántica específica asignada, y pueden ser referidas por definiciones de remitentes o por otras reescripciones establecidas.

La sintaxis de estos comandos es:

S_n

El comando S marca el comienzo de la regla e identifica a esta con un número. En la sintaxis S_n, n es el número de identificación de la regla.

Rlhs rhs comentario

El lhs es un formato aplicado a las entradas. Si este es igual, la entrada es reescrita a el rhs. Los comentarios son ignorados. Los campos tienen que ser separados por una etiqueta de carácter, estos pueden ser espacios agregados en el campo.

Las expansiones de macros de la forma \$x son ejecutadas cuando la configuración del archivo es leída. Las expansiones de la forma \$&x son ejecutadas en el tiempo que se usan, son un poco menos de lo que es un algoritmo general. Esto se hace sólo para referenciar macros definidas internamente tal como \$h que son cambiadas al ejecutarse.

El left-hand side (LHS)

El left-hand side de reglas de reescripción tiene un formato. Palabras normales que son simples de igualar directamente. La sintaxis de los metasímbolos inicia introduciendo un signo de pesos y se muestra en la tabla 3.1.

Tabla 3.1 Metasímbolos para la igualdad de formato

SÍMBOLO	SIGNIFICADO
\$*	igualdad nula o mas <i>tokens</i>
\$+	una igualdad o mas <i>tokens</i>
\$-	igualdad exacta en un <i>token</i>
\$=x	alguna igualdad en frases clase x
\$~x	alguna igualdad en palabra no perteneciente a clase x
\$x	igualdad en todos los <i>tokens</i> en la macro x

Los *tokens* son una cadena de caracteres en una dirección *e-mail* delimitada por un operador.

Si alguna de estas igualdades son asignadas a el símbolo \$n donde (n) es un número de *tokens*, esto reemplaza al formato LHS y sus *tokens* son asignadas al right-hand side (RHS), donde n es el índice en el LHS. Por ejemplo, supongamos la siguiente regla LHS:

\$-:\$+

Esta, supone la siguiente entrada:

UCBARPA:eric

En este ejemplo, la entrada de igualdad de la regla da otros valores con el RHS son:

\$1 UCBARPA

\$2 eric

Adicionalmente, el LHS puede incluir $\$@$ una igualdad de prueba nula. Esta no está destinada a $\$ N$ en el RHS y es solamente usada en igualdades de entrada nula.

Por ejemplo: $gceja@main.conacyt.mx$

Esta dirección e-mail tiene 7 token: $gceja$, $@$, $main$, $(.)$, $conacyt$, $(.)$, y mx . Esta dirección se formaría con un formato LHS de la siguiente forma:

$\$-@\$+$

La dirección especificada iguala el formato por lo siguiente:

•Hay exactamente un *token* antes de la $@$, la igualdad requiere un símbolo $\$-$.

•Contiene una arroba ($@$) que iguala a la de la dirección electrónica la cual es un *token* nulo.

•Hay un *token* o más después de la $@$, para la igualdad requiere un símbolo $\$+$.

El right-hand side (RHS)

Cuando en el LHS de una regla para reescribir de igualdad, la entrada es borrada y reemplazada por el RHS. Los *tokens* son copiados directamente de el RHS, si estos no comienzan con un signo de $\$$. Los metasímbolos utilizados se listan en la tabla 3.2.

Tabla 3.2 Metasímbolos para la transformación de direcciones de correo

SÍMBOLO	SIGNIFICADO
\$n	Substituye la identificación de token de LHS.
\$(name\$)	Especifica nombres canónico.
\$>n	Llama reglas establecidas n
\$#	Resuelve para el remitente
\$@	Especifica host
\$:	Especifica usuarios

El símbolo \$n substituye el valor correspondiente de un símbolo de igualdad \$+, \$-, \$*, \$=, o \$~ de el LHS. Este puede ser usado en cualquier lugar.

El \$(nombre\$) convierte un alias de host o a una dirección IP a un nombre canónico pasando por el valor del nombre a el nombre del servidor para la resolución.

El nombre de host es encerrado entre \$[y \$] es buscado usando el gethostbyname(SLIB) que son rutinas que lo reemplazan por el nombre canónico.

Los símbolos @\$ y \$: son usados para el control de procesos, los cuales previenen los loops. Si la transformación comienza con @\$, la entrada de la regla determinada es terminada y el resto que regresa de la transformación es el valor retornado por la regla determinada. Si la transformación comienza con \$: la regla es ejecutada sólo una vez. Se usa \$: para prevenir la recursión y prevé loops cuando son llamados por otras reglas. Se usa @\$ para salir de una regla determinada.

D define macro

La macros son definidas por medio de una letra mayúscula. Un nombre de macro puede ser definido con un solo carácter ASCII. Las letras minúsculas y los símbolos

especiales son usados internamente. El comando D define una macro y almacena un valor en esta. Una vez que la macro es definida, es usada para proporcionar el valor almacenado a otro comando del archivo sendmail.cf .

La sintaxis para la definición de macros es:

Dxval

La x es el nombre de la macro y val es el valor que esta tiene. Las macros son interpoladas usando la estructura \$x, donde x es el nombre de la macro a ser interpolada. Esta interpolación es realizada cuando la configuración del archivo es leída, excepto en líneas M. La estructura especial \$&x puede ser usada en líneas R para obtener una interpolación diferida.

Las condicionales pueden ser especificadas usando la siguiente sintaxis:

\$?x Esto es igual a la condicional IF(Si)

\$| Especifica else (entonces)

\$. Especifica endif

La macros listadas en la tabla 3.3 son definidas para usarse internamente por Sendmail para la interpolación.

Tabla 3.3 Macros internas para uso del comando D.

Nombre	Funcion
a	Formato original de fecha en RFC 822
b	Formato común de fecha en RFC822
c	Conteo de saltos
d	Formato de fecha en Unix (ctime)
e	Mensajes de entrada SMTP
f	Direcciones de envío (from)
g	Envío de direcciones relativas al receptor
h	Host receptor
i	Identificación de la cola
j	Nombre de dominio "oficial" para este sitio
l	Línea de formato de Unix
n	Nombre del <i>daemon</i> para mensajes de error
o	Establece operadores en direcciones
p	Identificación de procesos de Sendmail
q	Formato default para el envío de direcciones
r	Uso de protocolo
s	Nombre del host transmisor

Tabla 3.3 Continuación

t	Representación numérica del tiempo normal
u	Usuario receptor
v	Número de versión de Sendmail
w	Nombre del host del sitio
x	Nombre completo del transmisor
z	Directorio raíz del receptor

Existen tres tipos de fechas que se pueden usar. El **\$a** y **\$b** son macros del formato RFC 822; **\$a** es el tiempo que se extrae de la línea del mensaje de correo "Date" (si existe una), y **\$b** es la fecha normal y el tiempo (usado en el mensaje). Si la línea "Date" no se encuentra dentro del mensaje, la macro **\$a** determina el tiempo normal. La macro **\$d** es equivalente a la **\$b** en el formato del sistema Unix (ctime).

Las macros **\$w** y **\$j** determinan la identidad del host. El Sendmail trata de encontrar completamente el nombre calificado del host si esto es posible.

La macro **\$f** es el ID del transmisor determinado originalmente; cuando el correo especifica el host, la macro **\$g** determina la dirección del transmisor al receptor.

La macro **\$x** determina el nombre completo del transmisor. Esto puede determinarse de varias formas. La primera puede pasar como una bandera a Sendmail. La segunda escoge el valor de la línea "nombre-completo" en el encabezado si existe y la tercera escoge el campo comentario de la línea "From". Si lo anterior falla, y el mensaje existe localmente, el nombre completo es localizado en el archivo /etc./passwd.

Cuando se envía, las macros \$h, \$u y \$z establecen el host, usuario y directorio raíz (si es local) de el receptor. Las primeras dos son establecidas desde la \$@ y \$: parte de las reglas para reescribir respectivamente.

La macro \$c determina el "conteo de salto", esto es, el número en tiempo que el mensaje es procesado. Este se puede establecer por la bandera -h sobre la línea de comandos o por el conteo del timestamps en el mensaje.

Las macros \$r y \$s establecen el protocolo usado para la comunicación con Sendmail y el hostname del transmisor.

C y F define clase

Los comandos C y F definen clases de Sendmail. Una clase es un arreglo de valores. Las clases son usadas para cualquier cosa con múltiples valores que son manipuladas de igual forma, así como múltiples nombres para el host local o una lista de nombres de host UUCP

La sintaxis es:

C[clase][palabra1][...]

F[clase][nombre de archivo]

Estas clases usan símbolos especiales:

El símbolo \$= iguala algunos valores en esta clase.

El \$~ (igual a entradas no incluidas en esta clase) sólo iguala una palabra; multi-palabras que entran en la clase son ignoradas en este contexto.

Los valores de clase pueden ser definidos en una sola línea o en múltiples líneas, o cargados de un archivo, por ejemplo, la clase V se usa para definir conexiones UUCP. Para asignar una clase V a los valores gceja y saivarado se puede hacer en una sola línea de la siguiente forma:

CV saivarado gceja

Se establece que estas dos cuentas pueden manejar conexiones UUCP.

A diferencia de el comando C el comando F es utilizado para cargar valores de clase de un archivo. Este comando lee un archivo y almacena las palabras encontradas en una variable de clase. Por ejemplo, para definir una clase w y asignar todo esto a una cadena que se encuentra en el archivo /etc./sendmail.cw, se usa :

Fw/etc./sendmail.cw

M define remitente

El comando M define programas de distribución de correos usados por Sendmail. La sintaxis es:

M [nombre],[campo] = [valor]

El nombre es un nombre arbitrario usado internamente por Sendmail para referirse a este remitente y el par de campos "campo=nombre" define atributos para el remitente. Los campos manejados se especifican en la tabla 3.4.

Tabla 3.4 Definición de campos de remitente

Campo	Significado	Contenido
P	Path	Ruta del remitente
F	Flags	Banderas de Sendmail para el remitente
S	Sender	Reglas establecidas para el envío de direcciones
R	Recipient	Reglas establecidas para direcciones de destinatarios
A	Argv	Portador de argumentos de remitente
E	Eol	Fin de la línea de la cadena del remitente
M	Maxsize	Longitud máxima del mensaje

Las banderas usadas en los remitentes del campo F también son utilizadas para definir encabezados (H) y se muestran en la tabla 3.5

Tabla 3.5 Banderas usadas en el comando M y H.

Nombre	Función
C	Agrega una @ a direcciones que no tienen @
D	El remitente requiere un fecha: línea del encabezado
E	Agrega un > a las líneas del mensaje "From:"
e	Es un remitente costoso
F	El remitente requiere de un campo From: en la línea del encabezado
f	El remitente acepta una bandera -f de usuarios verdaderos

Tabla 3.5 Continuación

H	Mantiene letras mayúsculas en nombres de host
I	El remitente puede hablar SMTP hacia otro Sendmail
L	Límite de la longitud de la línea como es especificado en el RFC821
L	Este es un remitente local
M	El remitente requiere una identificación de mensaje en la línea de encabezado
M	El remitente puede mandar a múltiples usuarios en una transacción
n	No inserta el estilo Unix del campo From en línea del mensaje
p	El remitente requiere retorno de ruta en la línea
R	Usa el campo From regresando una ruta mejor que el retorno de direcciones
r	El remitente acepta una bandera -r de usuarios verdaderos
S	No establece la identificación del usuario antes de la llamada del remitente
s	Elimina las comillas de la dirección antes de la llamada del remitente
U	El remitente requiere el estilo Unix del campo From en la línea
u	Mantiene letras mayúsculas en los nombres de los usuarios
x	El remitente requiere el nombre completo en la línea del encabezado

H define encabezados

El formato de las líneas del encabezado que Sendmail inserta en los mensajes es definido por la línea H. La sintaxis de esta línea es la siguiente:

```
H[?bandera?][nombre]:[formato ]
```

El formato de la cabecera es una combinación de literales y macros incluidas en las líneas del encabezado.

La función de las banderas es muy simple. Las banderas del encabezado controlan si o no el encabezado es insertado dentro del correo destinado para un remitente específico. Si no hay banderas especificadas, el encabezado es usado para todos los remitentes. Si una bandera es especificada, el encabezado es usado sólo para un remitente que tiene la misma bandera establecida en la definición de remitentes. Las banderas del encabezado sólo controlan la inserción del encabezado. Si un encabezado es recibido en la entrada, este pasa a la salida sin tomar en cuenta las banderas establecidas.

O establece opciones.

El comando O asigna valores para opciones de Sendmail. Son opciones internas establecidas dentro del sendmail.cf. Hay un número de opciones "random" (de forma aleatoria) que pueden establecerse para la configuración del archivo. Las opciones son representadas por caracteres solos. La sintaxis es:

```
O[opción][valor]
```

El campo opción puede ser un valor de una cadena, un entero, un booleano (con valores falsos y verdaderos), o un intervalo de tiempo. La tabla 3.6 lista el significado y uso de cada opción.

Tabla 3.6. Opciones de Sendmail.

Nombre	Función
Afile	Define el nombre del archivo de alias.
aN	Espera N minutos para entrada @: @: y después reconstruye el archivo alias.
Bc	Define un carácter en blanco de substitución
D	Reconstruye la base de datos de alias
di	Distribución interactiva
dq	Distribuye durante la siguiente cola ejecutada
ee	Mensaje de error de correo y siempre retorna un cero de estado de salida
em	Regresa errores de mensaje de correo
ep	Impresión de mensajes de error
eq	Retorna sólo un estado de salida, y no mensajes de error
ew	Regresa mensajes de error escritos
Fn	Establece permisos para archivos temporales para n
f	Conserva un estilo de Unix en las líneas "From"
gn	Establece la identificación de grupo por default para remitentes de n.
Hfile	Define el nombre del archivo de ayuda SMTP
I	Usa el servidor de nombres BIND para resolver todos los nombres de host.
I	Ignora los puntos en una entrada de mensajes.
Ln	Establece el nivel del logging para n
Mxval	Establece una macro x para val

Tabla 3.6 Continuación

m	Envío de correo a sí mismo
Nnet	Define el nombre de la red local
o	Acepta formatos viejos de encabezado
Qdir	Define el nombre del directorio de cola
qn	Define un factor n usado para decidir cuando trabaja la cola
rt	Determina el intervalo t para un tiempo fuera de lectura
Sfile	Define el nombre de las estadísticas del archivo de servicio
s	Crea el archivo de cola antes de intentar distribuirlo
Tt	Establece el tiempo fuera de la cola para t
un	Establece la identificación del usuario por default para remitentes de n
v	Corre en modo extenso
wpass	Define el password usado para la ejecución de procesos paso a paso remotos.
Xl	Rechaza conexiones SMTP si se cargan promedios excedentes de l
xl	Mensaje de cola si se cargan promedios excedentes de l
Y	Distribuye cada cola de trabajo en procesos separados
yn	Baja prioridad de trabajos para n para cada receptor
Zn	Disminuye prioridades de trabajo para n cada vez que es ejecutado
zn	Factor usado con precedencia para determinar mensajes de prioridad.

P definición de Precedencias

Se utiliza para establecer prioridades de correo. Aquí, la elección es cuándo emplear prioridades. Ante el incremento de programas remitentes, personas externas que comienzan a usar prioridades de correo. Mientras más grande sea el número la prioridad es mayor.

El formato es: P[nombre de entrega]=[valor]

Por ejemplo, el programa de correo electrónico Eudora emplea las siguientes prioridades:

Highest (la más alta), High (alta), Normal (normal) Low (baja),Lowest (la más baja).
El listado siguiente muestra las definiciones:

Phighest=100
Phigh=50
Pnormal=0
Plow=-50
Plowest=-100

T Usuarios confiables.

La T define un usuario confiable; un usuario a quién se le permite enviar correo electrónico como otro usuario (un usuario normal sólo puede enviar correo desde su propia cuenta, mientras que un usuario confiable puede enviar correo desde cualquier cuenta). Normalmente los usuarios confiables estándares son root, daemon y UUCP.

La sintaxis es la siguiente:

T[usuario1][usuario2][...]

ejemplo: *Troot daemon UUCP*

3.4 Configuración de Sendmail.

La instalación del servidor Sendmail se lleva a cabo durante el proceso de configuración de Linux. Este incluye un archivo de configuración elemental Sendmail (*sendmail.cf*) el cual durante la instalación de Linux se configuró automáticamente, pero hay más trabajo que realizar para configurarlo de acuerdo a las necesidades particulares.

3.4.1 Configuración de Sendmail Localmente.

Para la configuración de entrega local si se permitió al programa de instalación Linux configurar el archivo *sendmail.cf* se realiza automáticamente. Si no fue así, se requiere que se instale este archivo de la serie n de Linux. Una vez que está instalado, hay que asegurarse de que algunos elementos se ejecuten con fluidez. Por ejemplo el *daemon*.

3.4.2 Ejecutando Sendmail como un *Daemon*.

Lo primero que se debe hacer es que el *daemon* Sendmail se inicie al momento del arranque. Un *daemon* (demonio) es un programa que se ejecuta en segundo plano y está pendiente de los elementos que maneja, para cuando se le solicita su servicio, en este caso el correo electrónico. Si no se tiene el *daemon* Sendmail en ejecución, no se detendrá ni esperará para procesar entradas y salidas de correo.

El *daemon* se carga normalmente en el directorio `/etc./rc.d` dentro del archivo `rc.net`.

El siguiente párrafo es un ejemplo de las líneas de código de Sendmail.

```
if [-f /usr/lib/sendmail ]; then
(cd /usr/spool/mqueue; rm -f if*)
/usr/lib/sendmail -bd -q1h: echo -n 'sendmail' > /dev/console
fi
```

Puede cambiar la localización o el nombre de archivo para configurar el *daemon*, pero el formato es muy similar en todas las distribuciones.

En el ejemplo anterior, la primera línea averigua la existencia del archivo `sendmail`.

La segunda línea cambia a su directorio de cola de correo, y suprime cualquier archivo bloqueado que podría haber quedado fuera, si tuvo que volver a arrancar mientras el sistema procesaba el correo.

La tercer línea es la que hace iniciar al *daemon* Sendmail. El modificador `-bd` indica al sistema ejecutar el *daemon* el cual utiliza el puerto 25. El modificador `-q1` indica que la cola se procese cada hora. Y lo restante de la línea sólo indica al sistema que coloque la palabra Sendmail en la pantalla.

Y la última línea es el indicador que concluye el enunciado `if/then`.

3.4.3 Como establecer Alias.

Para establecer los alias hay que editar el archivo `/etc./aliases`.

Se pueden agregar los alias que se deseen y se pueden colocar en el lugar que se desee, pero se recomienda ponerlos juntos y agregar líneas de comentarios con el fin de recordar para que se utilizan. Por ejemplo:

```
# Sendmail aliases
postmaster:root
admin:root
info:dee
rob:rjl
steven:stephen
```

Una vez que se guardo el archivo, al salir se debe de teclear el comando `newaliases` para activar los alias agregados.

3.4.4 Configuración de Sendmail para manejar Correo Internet.

Configurar correo electrónico para Internet es más complicado que procesar correo local. En esta parte se utilizan los comandos de configuración del archivo `sendmail.cf` vistos anteriormente.

El primer paso es identificar que estén definidas las siguientes líneas de configuración Internet:

La primera línea es para definir el host inteligente de relevo, indicándole a Sendmail que envíe cualquier cosa que no pueda manejar con sus propios grupos de reglas al servidor principal. El formato es el siguiente:

```
DSservidorprincipal.
```

Por ejemplo: vamos a configurar un servidor en una red de la compañía Seminarios, donde existen los siguientes equipos: `seminarioserv`, `seminario2`,

seminario3 y seminario4. El nombre de dominio de la compañía es unam.seminario.mx. Nuestro servidor principal será seminarioserv y las otras máquinas se dirigen por medio de él a Internet. Entonces en los equipos seminario2, seminario3 y seminario4, se colocaría en su correspondiente archivo sendmail.cf la línea siguiente:

DSe seminarioserv.unam.seminario.mx

Por otro lado el dominio de alimentación de la compañía Seminarios es cu.unam.mx. Por lo tanto en la máquina seminarioserv en su archivo sendmail.cf se coloca el nombre cu.unam.mx que es el proveedor del servicio de Internet:

DScu.unam.mx

Otro punto que hay que tomar en cuenta es a dónde se dirige el correo para nombres no calificados, indicándole a Sendmail a dónde enviar correo electrónico dirigido a alguien que no existe. En este caso las tres máquinas que no son el servidor principal de Seminarios, Usaría la siguiente línea:

DRseminarioserv.unam.seminario.mx

La siguiente línea es DHservidorprincipal. Esto determina hacia donde se dirige todo el tráfico local de correo electrónico. Si se emplea un servidor de correo centralizado, se puede utilizar el mismo nombre que hemos venido manejando. Pero si se utiliza un servidor de correo separado, se manejará el nombre de ese servidor en todas las máquinas.

La línea DMnombre.de.dominio. Se emplea para tener todo el correo electrónico enmascarado de un dominio, como si viniera desde un área común en particular. En

**ESTA TESIS NO DEBE
SALIR DE LA BIBLIOTECA**

todos los archivos `sendmail.cf` de todas las computadoras de la red Seminarios hay que colocar:

`DMunam.seminario.mx`

Esto significa que si alguien envía un correo electrónico a `usuario@dominio`, no será rechazado por qué olvido el nombre de host. Por ejemplo, la gente puede escribir un correo ya sea en `gonz@unam.semina.mx` o `gonz@seminario2.unam.seminario.mx`, y el mensaje llegará correctamente. Si no se tiene establecido el valor DM, las personas que envíen correo a `gonz@unam.semina.mx` tendrán su correspondencia de regreso, con un error de dirección no existente.

3.4.5 Sendmail y Correo UUCP.

La utilidad para redes más básica de Unix es una colección de comandos y tablas llamada UUCP (acrónimo de copia de Unix a Unix). UUCP es, de hecho, un comando de Unix.

Los sistemas conectados con UUCP se comunican a través de líneas de conexión serial, ya sea por medio de módem o de conexiones directas. No se requiere ningún software de red extra para ejecutar correo UUCP. El software es parte de la distribución de Linux.

La transferencia en estas conexiones se realiza en ambas direcciones. Por esta razón se puede tanto enviar como recibir correo electrónico. Los sistemas de correo electrónico usan tablas de reglas para revisar una dirección y elegir una forma de enviarla. De esta forma, un sistema de correo decide si conserva el correo en el sistema local, lo envía a un sistema por medio de una red UUCP o lo dirige a través de Internet.

Una dirección UUCP tiene la forma general *sitioremoto / destinatario*. La parte *sitioremoto* es el nombre de un sitio conocido por UUCP y *destinatario* puede ser un nombre de usuario o la forma general de una dirección UUCP. Para manejar correo UUCP no hay ningún cambio que se deba hacer siempre y cuando se emplee en el archivo `sendmail.cf` capacidad UUCP; este software se instala de la serie n de distribución de Linux. Sin embargo, sólo el servidor principal de correo debe tener la versión UUCP de este archivo. Los otros archivos `sendmail.cf` de las computadoras no deben tener capacidad UUCP.

3.4.6 Configuración de Sendmail para manejar Correo POP

Esta es una tarea sencilla, sólo hay que dar de alta el servicio de POP en el archivo `/etc/inetd.conf`. dentro de este archivo hay que agregar la siguiente línea (o sólo hay que borrar el signo # del principio de esta si ya existe en forma de comentario):

```
pop3 stream tcp nowait root /usr/sbin/tcpd /usr/sbin/in.pop3d
```

Si existe alguna línea similar que comience con `pop2`, hay que comentarla con el signo #, ya que es una versión anterior a la `pop3` del servidor `pop`.

Por último a las otras máquinas de la red en el archivo `inetd.conf` hay que comentar todo lo referente a `pop2` o `pop3`, ya que se requiere que sólo el servidor principal administre las conexiones `pop`.

3.5 Listas de correo.

Una lista de correo electrónico es una colección de direcciones personales electrónicas. Específicamente, el término se refiere al programa maestro que funciona en un Servidor de Correo Central, aceptando mensajes de dueños, moderadores, y miembros de listas. Los mensajes aceptados son adelantados

automáticamente a los demás miembros de la lista relevante. Las listas de correo electrónico usualmente tienen un tema o tópico de discusión específico. Estas son generalmente de tres tipos distintos: anuncios solamente, moderadas, o públicas, dependiendo de quien tiene permiso para enviar mensajes al grupo entero. Las listas de correo trabajan a base de un software llamado Majordomo, el cual hay que instalar.

¿Qué es Majordomo?

mayordomo: persona que habla, actúa, o está a cargo de los asuntos de otra.

Majordomo no comprende un sólo programa, sino mas bien una colección de rutinas escritas en lenguaje Perl usadas para automatizar la operación de una o más listas de correo electrónico. Majordomo se encarga de labores de rutina como son pedidos de suscripción y su cancelación. El Majordomo fue desarrollado originalmente por el Sr. Brent Chapman de GreatCircle Associates. Hoy en día, el Majordomo continúa siendo mejorado por Chan Wilson y el grupo de majordomo-workers.

A continuación se tratarán algunos consejos para la operación de listas de correo:

- El mensaje "Welcome to the list" (Bienvenido a la lista), tendrá una lista de políticas. Esta es muy similar a una política de uso aceptable, pues en ella se determina qué comportamiento es conveniente en la lista, el propósito de la misma y los temas aceptados.
- Se puede establecer que el majordomo coloque una pequeña línea en la parte inferior de cada mensaje de correo para señalar a la gente dónde obtener ayuda con la lista. Esto puede evitar problemas con quienes publican mensajes para solicitar ayuda.

- El majordomo viene con un archivo de ayuda estándar. Sin embargo, se puede personalizar para adaptarlo a nuestras necesidades.
- Es recomendable que sólo se moderen aquellas listas que necesitan serlo. Debido a que si se trata de modificar mucho las listas el moderador tendrá más trabajo que hacer.

3.6 Verificación del funcionamiento del correo electrónico.

El Sendmail proporciona poderosas herramientas para pruebas de configuración y la ejecución de procesos paso a paso para detectar errores. Estas herramientas son invocadas en la línea del comando Sendmail usando alguno de los argumentos de línea de comando Sendmail. Los argumentos se muestran en la tabla 3.7.

Tabla 3.7 Argumentos para ejecutar la línea de comandos de Sendmail.

Argumento	Función
-f addr	Envía direcciones de máquinas
-r addr	Forma obsoleta de -f
-h cnt	libera el correo si se envía en tiempos cnt
-F name	Determina el nombre completo del usuario para name
-n	No realiza alias o envía
-t	Envía cada lista en TO:, Cc:, Bcc:
-bm	Distribuye correo (default)
-ba	Corre en modo arpanet.
-bs	Habla SMTP del lado de la entrada
-bd	Corre como un <i>daemon</i>
-bt	Corre en modo de prueba
-bv	Verifica direcciones; no distribuye correo

Tabla 3.7 Continuación

-bi	Inicializa la base de datos de alias
-bp	Imprime la cola de correo
-bz	Congela la configuración del archivo.
-q[time]	Procesa la cola de correo. Lo repite en un intervalo de tiempo
-C file	Usa file como el archivo de configuración
-d level	Establece el nivel de debug
-o xvalue	Establece la opción x para la especificación de valor.
-e	Define como son retornados los errores.
-i	Ignora los puntos en la entrada de mensajes
-m	Envía correo a si mismo
-v	Corre en modo extendido
-s addr	Forma alternativa de -f

Algunos de los argumentos de línea de comandos son usados para verificar el procesamiento de direcciones para tener seguridad en la nueva configuración. Una vez que se cree que la configuración podrá trabajar, se eligen algunos sitios y direcciones de correo y se envían correos usando el argumento -C para leer la prueba de la configuración del archivo y el argumento -v para desplegar los detalles de la distribución de correo. -v despliega completamente el intercambio SMTP entre dos hosts.

CAPITULO 4
CASO PRACTICO

4. Caso Practico: Instalación y Configuración de un servidor de correo electrónico en LINUX dentro de la Facultad de Estudios Superiores Cuautitlán.

4.1 Introducción.

La Facultad de Estudios Superiores Cuautitlán (FES-C) cuenta con el servicio de correo electrónico, pero el equipo que lo ofrece no está centralizado. Por esta razón se instaló un servidor de correo electrónico maestro, el cual brinda una mejor administración en el correo electrónico tanto interno como externo, además de ofrecer mejor seguridad en el manejo de los mensajes, y por consecuencia proporciona al usuario un servicio óptimo y confiable.

En este capítulo se especifican los aspectos básicos que se llevaron a cabo para la instalación y configuración del servidor, como son la configuración de envío local y el envío a través de Internet, además de tocar aspectos de administración del sistema para un buen desempeño de éste.

4.2 Instalación y Configuración.

Como se comentó, para la instalación del servidor de correo electrónico, se debe contar con requerimientos de hardware y software (una distribución de Linux). El software elegido fue la distribución de Linux RedHat 4.2, por ser la versión más actual de Linux en esta distribución, y porque está disponible en el Centro de Cómputo de la FES-C. Por otra parte se requirió el siguiente equipo, basándonos en las características de hardware citadas en el capítulo 2:

CPU	486 DX2 66Mhz
BUS	ISA, EISA, PCI

RAM	16 MB de RAM.
Espacio en disco	1 GB en disco duro.
Monitor	VGA o SVGA.
Ratones	Un ratón estándar; Microsoft.
Unidad de CD-ROM	Esta unidad fue necesaria, dado que la distribución RedHat 4.2 está en ese formato. Asimismo se requirió de una tarjeta de red.

El primer paso para la instalación fue verificar que con la instalación previa de Linux se hayan cargado los programas referentes al correo electrónico. Verificamos entonces la existencia del archivo demonio Sendmail ubicado en `/usr/sbin/sendmail`.

Una vez que se comprobó la existencia del archivo `/usr/sbin/sendmail` se ejecutó éste archivo demonio desde el arranque del equipo, esto se hace editando el archivo `/etc/rc.d/sendmail.init`, donde se agregaron las siguientes líneas:

```
#!/bin/sh
#
# Este archivo se ejecuta al inicio y al final de una sesión de Linux, aquí se cargan
# y se establecen algunos archivos principales de configuración de sendmail.
#
# Establece bibliotecas de función
./etc/rc.d/init.d/functions
# Establece el archivo de configuración de la red.
./etc/sysconfig/network
# Checa que la red este operando correctamente y se cheque la existencia del
# archivo sendmail.
[ $(NETWORKING) = "no" ] && exit 0

[ -f /usr/sbin/sendmail ] || exit 0
```

En esta parte se llama al archivo /usr/sbin/sendmail para que se ejecute en segundo plano.

```
case "$1" in
start)
    # Inicio del demonio
    echo -n "Starting sendmail: "
    daemon sendmail -bd -q1h
    echo
    touch /var/lock/subsys/sendmail
    ;;
stop)
    # Cierre de sesión del demonio.
    echo -n "Shutting down sendmail: "
    killproc sendmail
    echo
    rm -f /var/lock/subsys/sendmail
    ;;
*)
    echo "Usage: sendmail {start|stop}"
    exit 1
esac

exit 0
```

Es recomendable incluir el archivo de inicio sendmail.init en un directorio rc, por ser el lugar donde se encuentran los archivos de arranque.

Después de cargar el programa demonio se establecen los alias en el archivo /etc/aliases de la siguiente forma:

Sistema básico de alias. En este se especifica al administrador del servidor de correo.

MAILER-DAEMON: postmaster

postmaster: jeolea

#

Redirecciones generales de pseudo cuentas.

bin: root

daemon: root

games: root

ingres: root

nobody: root

system: root

toor: root

uucp: root

#

Uso común de alias.

manager: root

dumper: root

operator: root

#

trap decode to catch security attacks

decode: root

#

Personas que pueden tener propiedad de root sobre el correo

#root: marc

zamed: hzamudio

jemendez: jeolea

Ya establecidos los alias, al salir hay que recordar teclear desde el prompt de Linux *newaliases* para que se establezcan los nuevos alias .

El siguiente paso es verificar la existencia del archivo principal de configuración */etc/sendmail.cf*. Con estos pasos el correo electrónico local quedó configurado. Ahora se prosigue con la configuración para manejar correo a través de Internet.

Para el servicio de correo a través de Internet, se agregaron las siguientes líneas en el archivo */etc/sendmail.cf*:

```
#####  
# Información local #  
#####  
# Se especifica el nombre del servidor maestro.  
Cwxe1-ha  
# archivo que contiene los nombres de cada cuenta que recibe correo.  
Fw/etc/mail/sendmail.cw  
# Se especifica el nombre de nuestro dominio.  
Dj$w.unam.mx  
# Define al host inteligente de relevo  
DS  
# se determina a dónde se dirige el correo para nombres no calificados  
DR  
# Determina hacia dónde se dirige todo el tráfico de correo electrónico  
DH  
#Se indica a Sendmail cuáles identificaciones de usuario no se pueden #enmascarar  
CE root  
# Se emplea esta línea para tener todo el correo enmascarado  
DM
```

Se especifica el nombre del demonio para mensajes de error.

DnMAILER-DAEMON

Finalmente para la configuración de correo POP se realizó lo siguiente:

Se verificó la existencia del programa demonio *in.pop3d* para el correo POP el cual está ubicado en el directorio */usr/sbin/in.pop3d*.

Después se editó el archivo */etc/inetd.conf*, en este archivo se especifican todos los servicios para usar con Internet. Se buscó la sección del archivo que permite el uso de servicios y se agregó la siguiente línea:

```
pop3 stream tcp nowait root /usr/sbin/tcpd /usr/sbin/in.pop3d
```

Esta línea habilita el servicio POP, se deshabilitó la línea similar *pop2*, debido a que *pop2* es una versión anterior al *pop3*.

En cada máquina cliente que recibe el servicio de correo electrónico del servidor maestro, se deshabilitó el servicio tanto UUCP como POP, ya que dicho servidor de correo es el único que debe manejar éstos. Esto se realizó comentando las líneas referentes a los servicios (UUCP, POP) dentro de los archivos */etc/inetd.conf* y */etc/services*.

Si no se encuentran los archivos expuestos anteriormente, se deben de instalar las series de red de Linux y configurar adecuadamente los datos de la red.

4.3 Puesta en marcha

Antes de probar el funcionamiento de nuestro servidor, se deben de crear algunas cuentas de correo electrónico, para el envío de mensajes.

Para la creación de cuentas se realizó lo siguiente:

Se edita el archivo `/etc/passwd` y se agrega una línea con los siguientes datos

1. Nombre del usuario o Login
2. Se deja un espacio para el campo clave o password
3. Número de identificación de usuario
4. Número de identificación de grupo.
5. Nombre completo del usuario.
6. Depto. al que pertenece u oficina. (opcional)
7. Teléfono (opcional)
8. Nombre del directorio hogar de la cuenta del usuario
9. Tipo de shell a utilizar.

El ejemplo de la línea para la cuenta es el siguiente:

```
hduarte:uwuwye:502:9:Moises Hernandez,Redes,6223415:/home/hduarte./sbin/bash
```

Después se copian los archivos de configuración de la cuenta del usuario ubicados en `/etc/skel` al directorio hogar de cada usuario. Luego se crea el directorio `/var/spool/mail`, el cual almacena los mensajes que reciben todos los usuarios.

Ahora que ya creamos algunas cuentas, se prosigue con el envío de mensajes entre estas cuentas para probar el correo local. Se utilizó el programa de correo electrónico *pine* incluido en el sistema operativo Linux.

Las pruebas fueron sencillas, se envía un correo a una cuenta con copia para diferente cuenta. Después se envía un mensaje a una cuenta con copia para nuestra propia cuenta para verificar que nuestra cuenta está recibiendo correo, y una última

prueba es enviar un correo a un grupo de correo electrónico especificado en los alias.

Por consiguiente, la prueba para envío de correo a través de Internet es similar, con la única diferencia que se deben de enviar correo a personas con cuentas fuera de nuestra red local, las cuales nos puedan contestar si es posible de forma inmediata.

4.4 Gestión del servidor de correo electrónico

Para la gestión se debe establecer un encargado de la administración del servidor y servicio de correo electrónico, el cual debe de realizar las tareas básicas de un operador de red y algunas específicas de correo como son:

- La creación de cuentas de correo
- Delimitación de espacio en disco.
- Restricciones de servicios.
- Mantenimiento del equipo del servidor
- Depuración de cuentas e información.
- Respaldo de información.
- Verificar el cumplimiento de las políticas
- Planes a futuro con respecto al crecimiento del servicio.

Para delimitar el espacio en disco se utiliza un programa demonio llamado *quotas*, el cual se encarga de establecer espacio en disco para cada cuenta de usuario, en este caso el espacio es de 2MB por cuenta.

También para un mejor desempeño del servidor y del servicio de correo es recomendable la restricción de algunos servicios como el FTP, Telnet y Finger, ya que el uso de estos aumenta la carga de trabajo del servidor, además por ser

servicios que permiten obtener información, normalmente rebasan el espacio del disco asignado a la cuenta.

El administrador debe hacer valer las políticas establecidas para evitar la existencia de información que no es válida, así como eliminar las cuentas que no son renovadas en el periodo establecido.

Con respecto al hardware del servidor, el administrador debe planear el cambio paulatino de componentes básicos como el aumento de capacidad de disco duro, de memoria y si es posible la actualización del procesador, para prever el crecimiento del servicio y la posibilidad de la instalación de un servidor secundario. Esto aumenta la seguridad y el desempeño del servicio.

CONCLUSIONES.

Primeramente, con el uso del sistema operativo Linux, bastó para la instalación y configuración del servidor de correo electrónico, se comprobó la eficiencia del Linux, la cual está bien respaldada por las características de Unix. El servidor de correo electrónico bajo Linux brinda un servicio confiable de igual o mejor forma que cualquier servidor de correo bajo un sistema operativo comercial como Windows NT.

Por otra parte, con respecto a la configuración del archivo `/etc/sendmail.cf`, es muy complicada por el uso de comandos especificados por una sola letra y el uso de signos manejados en la definición de macros internas.

La ventaja que se tiene en la configuración del archivo `/etc/sendmail.cf` es que con la instalación de Linux se configura una parte básica del servicio de correo automáticamente como es la configuración para reescribir direcciones de correo y manejar correo UUCP, pero aun así, no queda configurado completamente, debido a que se debe configurar a las necesidades de cada servicio.

Es muy importante que una vez configurado el servidor de correo, la administración se lleve adecuadamente, ya que de ahora en adelante se trabajarán sólo aspectos de administración y de ella depende el buen funcionamiento para brindar un servicio confiable para los usuarios. Dentro de estos puntos se debe considerar como uno de los más importantes el respaldo de información en un periodo de cada semana, para asegurar los mensajes de correo al enfrentarnos a problemas incontrolables en el servidor.

BIBLIOGRAFIA

1. Costales, Bryan. Sendmail. 2ª ed. E.U.: O'really, 1997. 1050 p.
2. LeBlanc Dee-Ann. Construye un site perfecto en internet con LINUX. México: Prentice Hall, 1998. 379 p.
3. Lamb, Linda. Using Email Effectively. E.U.: O'really, 1995. 160p.
4. eckett Jack, Jr. Linux. México: Prentice Hall, 1996. 831p.
5. E. Comer Douglas. Redes Globales de Información con Internet y TCP/IP. 3ª ed. México: Prentice Hall, 1996. 607 p.
6. Welsh, Matt. Runing LINUX 2ª ed. E.U.: O'really, 1996. 650p.
7. Feit, Sidnie. TCP/IP: Architecture, protocols and implementation. New York: McGraw Hill. ,1993. 466p (Computer communications).
8. Hunt, Craig. TCP/IP Network Administration. Sebastopol California: O'Reilly,1994. 472 p
9. Fiedler, David. UNIX System V. Indiana: Sams, 1996. 423p.
10. Dowd, Kevin. Getting Connected. E.U.: O'really, 1996. 424 p.
11. <http://sunsite.unc.edu/mdw/LDP/nag>.