UNIVERSIDAD NACIONAL AUTÓNOMA
DE MÉXICO

ESCUELA NACIONAL DE ESTUDIOS PROFESIONALES
ARAGÓN

INSTALACIÓN Y ADMINISTRACIÓN DE UN SERVIDOR WEB CON LINUX

T E S I S

QUE PARA OBTENER EL TÍTULO DE :
INGENIERO EN COMPUTACIÓN
P R E S E N T A :
LEOPOLDO MONROY JIMÉNEZ

ASESOR: ING. SERGIO A. ALVA ARGUINZONIZ

MÉXICO 2003

AGRADECIMIENTOS

A Dios:

Por brindarme la oportunidad de seguir siendo mejor cada día.

A mis padres:

Por mantener siempre constante su firma espíritu de lucha, por su profundo interés y confianza, logrando así poder disfrutar de este: "nuestro esfuerzo", por sus inmensos cuidados de noches de desvelo e inmensurable amor demostrado en todos los momentos de mi vida.

Por todo lo que para mi representan; mi infinito agradecimiento y eterna admiración.

A mi madre Rosa Jiménez Valenzuela: Por ser una gran amiga y una gran mujer, siempre inteligente, paciente, valiente, y fuerte, por ser tan maravillosa brindándome siempre su tiempo.

A mi padre Arturo Monroy Hernández: Por ser un gran amigo y un gran hombre inteligente, bueno, noble y emprendedor mostrando siempre su ejemplo de superación.

A mi hermana:

Angélica Monroy Jiménez: Por que vivimos juntos momentos dulces y amargos.

A mis abuelos:

Hermeregilda Valenzuela Cruz
Maria Hernández Alonso
Leonor Parra Sánchez
Leopoldo Monroy Parra
Remigio Jiménez Pérez
Roberto Monroy Parra

Por siempre creer en mi, y aunque algunos no se encuentren aqui, donde quiera que estén yo se que están conmigo.

A mi novia:

Haydeé Solup Mejia Ramírez: Por su inmensa capacidad de dar, su entrega incondicional, por estar conmigo en los momentos difíciles y compartir mis alegrías, por su inigualable apoyo y por abrirme las puertas de su casa en la cual encontré gente muy especial.

A mis amigos:

Por todos ustedes que de alguna manera me ayudaron.

AGRADECIMIENTOS

A mi asesor:

Ing. Sergio A. Alva Arguinzoniz: Por el apoyo, por su ayuda para concluir este trabajo y por la confianza que siempre tiene hacia los miembros del H. Departamento de Infraestructura del Centro Mascarones.

A mis profesores:

Por compartir sus enseñanzas y tiempo invertido para concluir satisfactoriamente este trabajo.

Índice

CAPÍTULO 1
GENERALIDADES

La distribución de Linux Slackware es una de las distribuciones más viejas que existen, desde el principio fue distribuida por medio de paquetes individuales identificados por una letra, esto hacia que fuera muy ligera y fácil de instalar en equipos viejos con poco espacio en disco duro.

Esta distribución fue elegida de entre otras como Red Hat Linux y LinuxPPP por su perfil, es una distribución que ocupa muy poco espacio, esta diseñada para montar un servidor, habilita los servicios necesarios para esta función, así mismo sus diseñadores han cuidado mucho mas el lado de la seguridad.

Otras distribuciones como Red Hat se han preocupado mas en diseñar una interfaz gráfica vistosa y han descuidado la seguridad. Enseguida se mencionan algunas características del sistema Linux para saber por que hoy en día es tan popular.

1. Historia.

Linux es un SO similar a UNIX, su historia comienza con una versión de UNIX para PC llamada Minix creada por el profesor Andrew Tannenbaum y que era utilizada para fines académicos.

Linux fue creado en el año de 1991, por un estudiante Finlandés de la Universidad de Helsinki llamado Linus Torvalds como proyecto escolar con la intención de crear una versión de Minix mejorada que se llamo Linux (contracción de Linus y UNIX).

Linux fue diseñado para trabajar específicamente con PC's basadas en microprocesadores X86. Desde su nacimiento Linux fue desarrollado y mantenido por grupos de programadores de todo el mundo, esto a raíz de que Linus distribuyo la primera versión de Linux en forma gratuita por Internet con todo y su código fuente, invitando a participar a todo aquel que tuviera interés en el tema y los conocimientos de programación necesarios para examinar el código y detectar errores o añadir mejoras al mismo.

Hoy en día Linux sigue siendo mantenido por una gran comunidad de personas del todo el mundo no tan solo programadores, sino también gente común y corriente. Después de 10 años Linux a logrado ganar presencia dentro del ambiente de los SO.

1.2 Características

1.2.1 Diseño

Linux es un SO robusto y confiable, por ser un sistema parecido a UNIX ofrece las misma ventajas que este, y no requiere de un costoso hardware para su funcionamiento.

Por estas características Linux se ha convertido en una buena alternativa tanto para empresas pequeñas como para centros educativos.

Algunas de las caracteristicas que Linux comparte con UNIX son:

- **Multitarea:** permite realizar varias tareas a la vez.
- **Multiusuario:** Linux admite sesiones multiusuario, lo que implica que varios usuarios pueden acceder a un servidor Linux simultáneamente.
- **Sistema jerárquico de archivos:** Linux tiene un sistema jerárquico de archivos. Su directorio superior contiene subdirectorios que se subdividen en otros subdirectorios. Juntos, estos subdirectorios forman una estructura de árbol.
- **Memoria virtual:** algunas veces se necesita acceder a mas memoria RAM de la que se tiene físicamente, en la computadora. La memoria virtual es un espacio en disco duro que se comporta como memoria RAM extra cuando es requerida por el sistema.
- **Librerías compartidas:** para crear programas en Linux no es necesario que el programador cree sus propias rutinas, si necesita de una rutina en especial puede utilizar las librerías compartidas que posee Linux, de esta manera podemos tener programas más pequeños y por consiguiente ahorro tanto de disco duro como de memoria RAM.
- **Carga por demanda:** cuando un proceso se encuentra activo, pero no en uso por algún tiempo, un sistema como Linux lo moverá al espacio de memoria virtual con el fin de liberar espacio en RAM física para las tareas que se están ejecutando en este momento.
- **Administración apropiada de memoria:** un aspecto muy importante en un SO es la administración apropiada de memoria, si un SO no tiene un apropiado uso de la memoria entonces este se vera con bloqueos constantes de programas y por consecuencia el reinicio del sistema, Linux solo asigna lo necesario para que cada programa se ejecute sin tener que reiniciar el sistema si uno de estos procesos se bloquea.
- **Redes TCP-IP:** Linux cuenta con los paquetes TCP-IP para poder conectarse a una red, además cuenta con todos los programas necesarios para conexión de redes.
- **POSIX:** Portable Operating System Interface, puesto en marcha por el Instituto Nacional de Estándares y Tecnología de los Estados Unidos. Linux cuenta con este estándar, esto quiere decir que las aplicaciones para otras versiones de UNIX se transportan a Linux sin mucha dificultad.

Un gran numero de aplicaciones de UNIX se han pasado a Linux por consiguiente, la forma de utilizarlo y el aspecto de Linux es muy similar al de UNIX.

1.2.2 Filosofía

Linux es considerado un clon de UNIX sin serlo completamente, mientras que UNIX se distribuye bajo una licencia comercial, Linux se distribuye bajo la GNU Public License (Licencia Publica GNU).

Algunas de las características de licencia GPL son:

Permite que Linux sea distribuido en forma libre, sin importar el medio de distribución ya sea por medio de Internet, Revistas, Mirrors, Libros o versiones comerciales.

También especifica que Linux debe ser distribuido con su Kernel completo incluyendo su código fuente.

Al igual que el Kernel, los demás programas deben contener su código fuente para poder ser modificados si así lo desea el usuario, pero respetando los derechos de autor.

Esta licencia es un tanto complicada pero se puede obtener mas información sobre ella en la pagina www.gnu.org.

Muchos programas como Linux son considerados Open Source o de código abierto, esto implica desarrollar software de calidad para poder ser utilizado como software de producción en empresas, pero con un bajo costo.

1.3 Plataformas y distribuciones más comunes.

A pesar de que la primera versión de Linux solo trabajaba con procesadores Intel hoy en día Linux puede ejecutarse en casi cualquier hardware, entre los que se encuentra:

- Procesadores AMD y Cyrix.
- Procesadores Alpha de Digital.
- Procesadores 80386, 80486, 80586 y Pentium Intel.
- Procesadores AMD-K6, Celeron y Athon
- Procesadores Power PC de Macintosh.
- Procesadores Sparc.

Existen diferentes distribuciones de Linux algunas comerciales y otras gratuitas, pero todas ellas se distribuyen bajo la licencia GNU. Algunas de estas versiones son:

Slackware (www.slackware.com).
RedHat (www.redhat.es).
Suse (www.suse.com).
Debian (www.debina.org).

Sin importar la distribución todas los Linux contienen el mismo Kernel. El Kernel es la parte medular de un SO sin el ningún SO funcionaria, este Kernel pasa por varias

pruebas de calidad y después es puesto en Internet para que todos puedan obtener la nueva versión (www.kernel.org).

Cada distribución tiene su propia forma de ser instalada ya sea con una vistosa interfaz gráfica o tan solo con ventanas en modo texto. Para este documento se utilizara la distribución de Linux Slackware.

1.4 Aplicaciones.

Linux utiliza los mismos comandos que un sistema UNIX, maneja la misma sintaxis y las mismas opciones. El manejo de comandos en Linux es muy similar al uso de comandos en MS-DOS.

Linux proporciona varias utilidades para el manejo de archivos, dispositivos y la administración del sistema como son:

- Editores de textos.
- Interfaces gráficas.
- Herramientas de manejo de dispositivos (CD-ROM, floppy, etc.).
- Herramientas de administración de archivos.
- Herramientas de administración del sistema (cron. at, etc.).
- Herramientas para la administración del sistema.

1.4.1 Procesamiento de palabras y procesamiento de texto.

Linux proporciona un gran número de programas para el procesamiento de palabras y texto, algunos de ellos los podemos emplear directamente en la línea de comandos, o solamente escribiendo el nombre de este en la línea de comandos. O bien si contamos con una interfaz gráfica podemos utilizar los que vienen incluidos en ellas.

Algunos de estos procesadores cuentan con una interfaz muy similar a las de Office, otras simplemente utilizan una ventana en modo texto y hacen uso de combinaciones de teclas para la ejecución de ciertas tareas.

Ya sea en modo texto o con una interfaz gráfica estas aplicaciones son altamente efectivas. Algunos de estos procesadores son:

- Vi
- Emacs
- Ed
- Openoffice

1.4.2 Lenguajes y utilerías de programación.

Linux ofrece un gran numero de lenguajes y utilerias de programación entre las que se incluyen:

C, un lenguaje de alto nivel la mayor parte del kernel de Linux esta escrito en C.

Shell, es un programa que se ejecuta cuando abrimos una sesión en Linux, que además de ayudarnos a interpretar comandos también funciona como un lenguaje de programación.

Perl, un lenguaje de scripts de propósito general que suele utilizarse para el desarrollo de CGI'S.

Phyton, un lenguaje de scripts transportable, interpretado y orientado a objetos que comparte muchas de las características de Perl, Tcl y Java.

PHP, un lenguaje de programación para el tratamiento de información en paginas web.

1.4.3 El sistema X Window.

El sistema X-Windows es un entorno gráfico desarrollado por el Tecnológico de Massachusetts, que fue desarrollado primero para sistemas UNIX y después fue pasado a Linux.

Este sistema X-Windows funciona mediante una relación cliente/servidor. Esto quiere decir que el sistema X-Windows utiliza un Servidor de ventanas X que es el encargado de dibujar las ventanas en pantalla y de administrar los recursos (Monitor y tarjeta de video) para que esto pueda ser posible. El encargado de mostrarnos un escritorio con sus iconos, su barra de tareas, etc. Se le conoce con el nombre de gestor de ventanas (KDE, GNOME, etc.) que es el programa cliente.

El Servidor de ventanas X solo se encarga de dibujar en el monitor lo que el gestor le indica, si el gestor necesita una ventana en una posición y un color determinado simplemente el gestor le pasa estos datos al Servidor y el se encarga del resto.

Existen varios gestores de ventanas para Linux, entre los mas conocidos tenemos a KDE y GNOME por mencionar algunos, estos dos gestores de ventanas son los más populares dentro del ambiente Linux que tratan de hacer más agradable el trabajo para el usuario.

1.4.4 Redes y comunicaciones.

Linux es una excelente plataforma para montar un servidor tanto para Internet como para una Intranet, ya que ofrece una optima potencia a las redes y proporciona clientes y servidores para todos los protocolos esenciales, entre los que se incluye:

- **FTP** (Protocolo de Transferencia de Archivos).
- **HTTP** (Protocolo de Transferencia de Hipertexto).
- **IP** (Protocolo de Internet).
- **NNTP** (Protocolo de Transferencia de Noticias en Red).
- **POP** (Protocolo de Oficinas de Correo).
- **PPP** (Protocolo Punto a Punto)
- **SLIP** (Protocolo de Internet por Linea Serie).
- **SMTP** (Protocolo Simple de Transferencia de Correo).
- **TCP** (Protocolo de Control de Transmisión).

Linux es con toda seguridad el sistema operativo más optimizado para redes existente en la actualidad, Incluso llega a admitir protocolos de red de otros sistemas operativos, incluyendo Windows 98/NT/2000, Novell y MacOS . De este modo los servidores Linux se integran perfectamente en cualquier entorno heterogéneo.

1.4.5 World Wide Web.

Actualmente Linux es el SO que se encuentra instalado en el 50% (http://www.netcraft.com/survey/ noviembre del 2002) de los servidores de Internet.

Linux cuenta con diferentes herramientas de desarrollo para Internet como son lenguajes de programación y un software especial para poder instalar un servidor de web (apache).

Apache actualmente es el mejor software para servidores Web, y esto es debido a su diseño, ya que después de ser instalado puede ser personalizado de acuerdo con las necesidades de cada usuario.

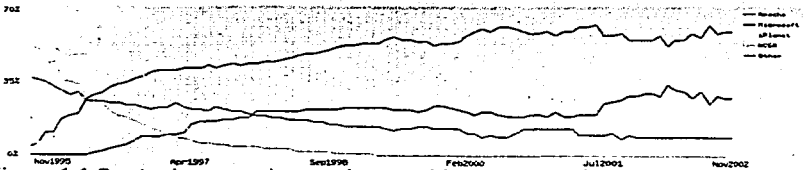


Figura 1.1 Cuota de mercado para los servidores agosto 1995 - noviembre 2002. (http://www.netcarft.com/Survey/index-200211.html)

Apache se puede utilizar también como servidor proxy, aunque este no es su fuerte, bien puede servir para poder proveer de Internet a una pequeña Intranet.

Con estas características Linux se postula en un futuro como el SO que predomine en los servidores de Internet.

1.4.6 Otras Aplicaciones.

Linux aparte de lo ya mencionado pude aplicarse en otras áreas por ejemplo:

- Se puede utilizar como una estación de trabajo.
- Servidor de impresión.
- Servidor de chat con el software de IRC (Internet Relay Chat).
- Servidor proxy con squid
- Servidor ftp.
- Firewall.
- Servidor Telnet.
- Servidor de correo.

Por mencionar algunas aplicaciones. Linux es muy flexible y puede ser utiliza para una gran variedad de aplicaciones es por eso que hoy en día Linux cuenta con bastantes adeptos y se puede considerar un fuerte contrincante de los sistemas Windows.

CAPÍTULO 2
INSTALACIÓN

En este capítulo se busca describir como instalar un servidor web donde se alojará la página principal del Centro Mascarones encontrar la solución acorde a las necesidades del Centro buscando la mejor combinación costo beneficio para la elección del equipo y la plataforma.

Con la experiencia obtenida en este proceso de aprendizaje se busca que ésta tesis sirva como apoyo a aquellas personas que buscan una alternativa viable en el software libre para instalar servidores de alta calidad, gran desempeño y confiabilidad.

2 Instalación.

La mayoría de las distribuciones de Linux se distribuyen en uno o varios CD-ROM, esto implica una gran ventaja ya que todo lo que necesita Linux para funcionar se incluye dentro del CD-ROM.

Para algunos usuarios novatos instalar Linux no resulta algo trivial y cuando por fin logran instalarlo lo hacen de forma completa con todos los paquetes que se incluyen en el CD-ROM.

Esto no suele ser un problema cuando se trata de un equipo casero, pero cuando hablamos de un equipo que funcionara como servidor puede resultar no una buena idea instalar paquetes que no serán utilizados y sólo estarán ocupando espacio en disco duro, que podría ser utilizado para descargar e instalar paquetes mucho mas útiles para las necesidades del servidor que algunas que vienen incluidas en los discos de instalación.

El CD-ROM de la distribución Linux Slackware versión 7.1 contiene los paquetes necesarios para una instalación completa, pero se omitirán algunos de estos paquetes y sólo se seleccionaran aquellos que vayan de acuerdo con lo que se ésta buscando, que es hacer funcionar a Linux como un servidor de paginas web.

2.1 Requerimientos.

Slackware Linux no requiere de un sistema extremadamente potente para ejecutarse.

Linux Slackware se puede ejecutar en equipos 386 o superiores. Los requerimientos mínimos para instalar y ejecutar Slackware son:

- Procesador 386.
- 16MB en RAM.
- 50 mega bites de espacio libre en disco duro.
- Unidad de 3,5".
- Tarjeta de red.

El recomendado basándose en la experiencia de este proyecto es:

- Procesador Pentium.
- 128 en RAM.
- 2000 mega bytes de espacio libre en disco duro.
- Unidad de CD-ROM.
- Unidad de 3.5".
- Tarjeta de red.
- Tarjeta de video.

2.2 Hardware del servidor

La máquina que funcionará como servidor de páginas web, tiene las siguientes características.

- Procesador Intel Celeron Pentium III
- 128RAM.
- 20 giga bytes en disco duro.
- Unidad de CD-ROM 52X.
- Unidad de 3.5".
- Tarjeta de red intel ethernet pro 100.

Es necesario mencionar que para instalar Slackware se debe seleccionar las series de paquetes que se instalarán.

2.3 Series de paquetes.

Linux Slackware es una de las distribuciones de Linux más antiguas, contiene un conjunto de paquetes que se reparten en series, cada una de estas series ésta etiquetada con una letra del abecedario comenzando con la letra A hasta llegar a la Z.

La forma tan particular que Slackware tiene para manejar los paquetes formo un estándar entre las diferentes distribuciones Linux.

Era mucho más rápido conseguir una distribución de Linux Slackware de ésta manera que tener que descargar una distribución completa que incluía paquetes que tal vez nunca se utilizaban.

Esta característica hace que Slackware sea mucho más flexible que otros Linux permite personalizar la instalación de acuerdo con las características del equipo y la tarea que se le será asignada como proporcionar algún tipo de servicio (telnet, ftp, http, etc.).

No es necesario descargar toda la distribución completa de Linux Slackware para que ésta funcione sólo se debe seleccionar los paquetes indicados para poder lograr que ésta trabaje como uno desea.

Esto no es una tarea fácil, se necesita de mucha paciencia y experiencia para saber que paquetes son los necesarios para un determinado tipo de instalación, ésta experiencia se obtiene cuando se ha realizado varias veces la instalación y se sabe con cierta claridad que incluye cada serie de paquetes y si ésta puede ser omitida sin afectar el funcionamiento del sistema cuando ya ésta instalado.

Se debe probar varias veces la instalación realizando una combinación diferente de paquetes en una maquina que no sea el servidor pero con similares características y que funcione para este fin, una máquina de pruebas. De ésta manera se puede adecuar la instalación para saber cual se adapta mas a lo que se busca.

No existe una fórmula que diga como seleccionar los paquetes correctos para instalar un servidor web, pero se pueden seguir algunas reglas que pueden ayudar para este fin. Por ejemplo se puede omitir los paquetes de la serie X si no se desea trabajar con X Windows.

La serie A contiene los suficientes programas para ejecutar Linux si se omite ésta serie de paquetes el sistema no funcionara, existen otras series que se pueden omitir por completo como la serie U que son programas que sólo trabajan en sistemas UltraSparc. Estos son sólo algunos paquetes que se pueden agregar u omitir pero existen mas, a continuación se escribe una lista con los nombres de las series y sus características.

A - es la base que contiene suficiente software para levantar y ejecutar Slackware, contiene algunos editores de texto y programas de comunicaciones.

AP - varias aplicaciones que no requieren del sistema X Windows.

D - Herramientas de desarrollo de programas, compiladores, depuradores, intérpretes, y los manuales man.

DES - Incluye la función de crypt() de libc de GNU.

E - Gnu Emacs, es tan grande que requiere su propia serie.

F - Contiene FAQs, HOWTOs, y otro tipo de documentación.

GTK - Contiene el ambiente de escritorio de GNOME, biblioteca de widget de GTK, y el GIMP.

K - El código fuente para el núcleo de Linux.

KDE - Contiene el ambiente de trabajo de escritorio KDE.

N - Contiene programas para configuración de una red. Demonios, programas de correo, telnet, programas de lectura de noticias, etcétera.

T - Contiene el sistema de formato de documentos teTeX.

TCL - Contiene las herramientas del lenguaje de comandos, el Tk, el TclX, y el TkDesk.

U - Contiene paquetes de programas diseñados especificamente para trabajar solamente en sistemas de UltraSPARC.

X - Contiene la base del sistema X Window.

XAP - Contiene aplicaciones X que no son parte de un ambiente de escritorio importante. Por ejemplo Ghostscript y Netscape.

XD - Contiene Bibliotecas, kit de la conexión del servidor, y ayuda de PEX. Para el desarrollo de X11.

XV - Contiene Bibliotecas de XView, manejadores de ventanas aplicaciones de otras de XView.

Y - Contiene juegos (una colección de juegos de BSD).

La decisión que se tomó después de tomar en cuenta las características antes mencionada y de acuerdo con la experiencia acumulada después de haber instalado en varias ocasiones Linux Slackware, fue de la lista de paquetes que se incluyen en la distribución sólo seleccionar los paquetes de la serie A, D, F, K y N, con esto el sistema sólo ocupara alrededor de 500MB y se minimiza la cantidad de paquetes que no se utilizan, habrá espacio suficiente para instalar paquetes extras que no se incluyen en el CD-ROM de la distribución y que son necesarios para el servidor.

2.4 Inicio de la instalación.

Después de haber elegido los paquetes se procede a iniciar la instalación de Linux Slackware, basta con insertar el CD de la distribución en la unidad de CD-ROM y reiniciar el equipo.

Cuando el sistema haya arrancado nuevamente se ejecutara la primera parte de la instalación de Linux se mostrara una pantalla de bienvenida al comienzo de la instalación donde aparecerá en la parte inferior de ésta pantalla el indicador boot:.

```
SYSLINUX 1.48 1999-09-26  Copyright (C) 1994-1999 H. Peter Anvin

Welcome to Slackware version 7.1.0 running Linux version 2.2.16!

If you need to pass extra parameters to the kernel, enter them at the prompt
below after the name of the kernel to boot (vmlinuz).  NOTE:  In most cases the
kernel will detect your hardware, and parameters are not needed.

Here are some examples (and more can be found in the BOOTING file):
   hdx=cyls,heads,sects,wpcom,irq  (needed in rare cases where probing fails)
or hdx=cdrom (force detection of an IDE/ATAPI CD-ROM drive) where hdx can be
any of hda through hdh.

In a pinch, you can boot your system from here with a command like:
   vmlinuz root=/dev/hda1 load_ramdisk=0 initrd=

This prompt is just for entering extra parameters.  If you don't need to enter
any parameters, hit ENTER to continue.

boot: _
```

Figura 2.1 Pantalla de bienvenida y preparación de la instalación.

Esta pantalla contiene información sobre algunos parámetros extras que se pueden pasar durante el arranque del sistema, como no se necesita pasar ningún parámetro extra al núcleo sólo se pulsa enter y de ésta forma se iniciara la primera parte de la instalación.

En caso de que no se haya elegido ninguna opción en un determinado tiempo (1 minuto) automáticamente se lanzara el programa de instalación.

Linux Slackware detectará automáticamente el hardware que ésta instalado en el equipo.

```
Memory: 28368k/32768k available (1688k kernel code, 412k reserved, 792k data, 11
6k init)
Dentry hash table entries: 4096 (order 3, 32k)
Buffer cache hash table entries: 32768 (order 5, 128k)
Page cache hash table entries: 8192 (order 3, 32k)
VFS: Diskquotas version dquot_6.4.0 initialized
CPU: L1 I Cache: 32K  L1 D Cache: 32K
Enabling new style K6 write allocation for 32 Mb
CPU: AMD AMD-K6(tm) 3D processor stepping 8c
Checking 386/387 coupling... OK, FPU using exception 16 error reporting.
Checking 'hlt' instruction... OK.
Checking for popad bug... OK.
POSIX conformance testing by UNIFIX
mtrr: v1.35a (19990819) Richard Gooch (rgooch@atnf.csiro.au)
PCI: PCI BIOS revision 2.10 entry at 0xfd900
PCI: Using configuration type 1
PCI: Probing PCI hardware
PCI: Enabling memory for device 00:00
Linux NET4.0 for Linux 2.2
Based upon Swansea University Computer Society NET3.039
NET4: Unix domain sockets 1.0 for Linux NET4.0.
NET4: Linux TCP/IP 1.0 for NET4.0
IP Protocols: ICMP, UDP, TCP, IGMP
TCP: Hash tables configured (ehash 32768 bhash 32768)
_
```


Figura 2.2 Detección de hardware.

Una vez finalizada la detección del hardware aparecerá la pantalla de bienvenida a la instalación de Linux Slackware e iniciara la segunda parte del proceso de instalación.

Antes de continuar con la segunda parte de la instalación se necesita agregar una partición de tipo nativa y una partición de tipo swap, para poder realizar modificaciones dentro del sistema se necesita tener permisos de superusuario (root), para poder obtener estos permisos se escribe root y se oprime enter.

```
Welcome to the Slackware Linux bootable installation CD! (version 7.1.0)

******  IMPORTANT!  READ THE INFORMATION BELOW CAREFULLY.  ******

- You will need one or more partitions of type 'Linux native' prepared.  It is
  also recommended that you create a swap partition (type 'Linux swap') prior
  to installation.  For more information, run 'setup' and read the help file.

- If you're having problems that you think might be related to low memory (this
  is possible on machines with 8 or less megabytes of system memory), you can
  try activating a swap partition before you run setup.  After making a swap
  partition (type 82) with cfdisk or fdisk, activate it like this:
    mkswap /dev/<partition> ; swapon /dev/<partition>

- Once you have prepared the disk partitions for Linux, type 'setup' to begin
  the installation process.

- If you do not have a color monitor, type:  TERM=vt100
  before you start 'setup'.

You may now login as 'root'.

slackware login: _
```

Figura 2.3 Pantalla de acceso al sistema para la instalación.

Linux cuenta con su propio fdisk, para poder crear particiones de tipo nativa y swap en el disco duro, ésta herramienta es parecida al fdisk de Windows pero con muchas mas y mejores caracteristicas.

Linux tiene su propia forma de nombrar a los diferentes dispositivos que se encuentran instalados en la computadora, como son: discos duros, discos flexibles, unidades de CD-ROM, etcétera.

Los discos duros en especial son nombrados de acuerdo a su interfaz de conexión y al número de particiones.

Al contrario de lo que sucede en Ms-Dos o Windows, en donde los dispositivos tales como discos flexibles, unidades de CD-ROM, discos duros y sus particiones se nombran con una letra del abecedario C:, D:, E:, etc., sin importar el tipo de interfaz de conexión, ésta es tal vez la forma en como los usuarios reconocen su disco duro.

Para que sea más fácil ubicar la forma en como Linux reconoce a los discos duros, enseguida se muestra una tabla comparativa entre Windows y Linux, y la forma en como reconoce cada sistema los discos duros:

Linux	Descripción
/dev/hda	Disco duro ide primario (maestro).
/dev/sda	Disco duro scsi primario (maestro).
Windows	Descripción
C:	Disco duro ide primario (maestro).

Tabla 2.1 Discos duros en Linux y Windows.

Para reconocer discos esclavos, entre Linux y windows se muestra a continuación una tabla comparativa.

Linux	Descripción
/dev/hdb	Disco duro ide secundario (esclavo).
dev/sdb	Disco duro SCSI secundario (esclavo).
Windows	Descripción
D:	Disco duro ide secundario (esclavo).

Tabla2.2 Discos duros esclavos en Linux y Windows.

La forma en como reconoce Linux y Windows una partición de describe a continuación en la siguiente tabla.

Linux	Descripción
/dev/hda1	Disco duro ide primera partición (maestro).
/dev/sda1	Disco duro scsi primera partición (maestro).
Windows	Descripción
C:,D:,E:,F:,etcétera.	Disco duro ide primera partición (maestro).

Tabla 2.3 Particiones en Linux y Windows.

Como el servidor sólo cuenta con un disco duro, la forma de ejecutar fdisk es la siguiente:

#fdisk /dev/hda

```
Linux 2.2.16.

If you're upgrading an existing Slackware system, you might want to
remove old packages before you run 'setup' to install the new ones.  If
you don't, your system will still work but there might be some old files
left laying around on your drive.

Just mount your Linux partitions under /mnt and type 'pkgtool'.  If you
don't know how to mount your partitions, type 'pkgtool' and it will tell
you how it's done.

To partition your hard drive(s), use 'cfdisk' or 'fdisk'.
To activate PCMCIA/Cardbus devices needed for installation, type 'pcmcia'.
To activate network devices needed for installation, type 'network'.
To start the main installation, type 'setup'.

# # fdisk

Usage: fdisk [-l] [-b SSZ] [-u] device
E.g.: fdisk /dev/hda  (for the first IDE disk)
  or: fdisk /dev/sdc  (for the third SCSI disk)
  or: fdisk /dev/eda  (for the first PS/2 ESDI drive)
  or: fdisk /dev/rd/c0d0  or: fdisk /dev/ida/c0d0  (for RAID devices)
  ...
# _
```

Figura 2.4 Acceso a fdisk.

Una parte importante en la creación de las particiones es como será distribuido el espacio total en disco, es muy común encontrar discos duros con dos particiones la partición nativa y la de swap.

Muchos usuarios de Linux se sienten satisfechos si pueden finalizar la instalación sin problemas, por lo que son propensos a evitar esquemas de particiones mas complicados.

Existen pocas rutinas de instalación que resalte la relación entre particiones y seguridad. No indican que dichas configuraciones conllevan ciertos riesgos.

No se debe colocar el sistema de archivos raíz y de usuario en la misma partición Linux. Esto aumenta la posibilidad de que las personas que deseen realizar ataques puedan explotar los programas SUID para acceder a áreas restringidas.

Tener en una misma partición todo el Linux dificulta la tarea de los administradores de sistemas. Puede dificultar la capacidad para actualizar o hacer copias de seguridad. Incluso algunos archivos dañados pueden provocar problemas, lo que significa que la jerarquía de un directorio dañado puede afectar a los réstantes, incluso puede obligar a reinstalar Linux.

Para evitar estos problemas, se debe crear una partición independiente para cada uno de los sistemas principales de archivos.

Dicha configuración mejora la seguridad y permite gestionar las copias de seguridad y su posterior recuperación. La existencia de varias particiones ofrece varias ventajas por ejemplo:

- Una sencilla gestión de las copias de seguridad y actualizaciones.
- Arranque más rápido (en algunos casos).
- La capacidad para controlar como se monta cada sistema de archivos.
- Otra ventaja mas es que evita la denegación de servicio accidental y protege de desbordamientos al sistema de archivos raiz.

Como se ha indicado, los usuarios nuevos tienden a huir de la creación de varias particiones, debido a que se debe tomar algunas decisiones importantes, como por ejemplo el tamaño que debe tener cada una de las particiones y donde serán montadas cada una de ellas.

El termino montar hace referencia a la forma en que Linux permite utilizar los distintos sistemas de archivos.

Cuando Linux monta un sistema de archivos local o externo, conecta el sistema a un dispositivo o directorio local, lo que permite un punto de acceso o punto de montaje.

A excepción de las particiones de raiz y swap, el tamaño y el punto de montaje de algunas particiones dependen de distintos factores.

Uno de esos factores es la función que va a tener Linux. Por ejemplo, si va a haber muchos usuarios, o si va a proporcionar algún tipo de servicio como correo electrónico, paginas Web, etcétera.

Algo que puede ayudarnos es conocer las funciones de los distintos sistemas de archivos. Dichas funciones son:

- /:contiene relativamente pocos archivos (sobre todo scripts de inicio).
- /usr: contiene la mayoría del software.
- /home: contiene los directorios de los usuarios.
- /opt: contiene el software complementario de terceros (Netscape, StarOffice, etcétera.).
- /var: contiene registros administrativos, correo y noticias.

Los parámetros de las particiones anteriores sólo se han utilizado para facilitar las explicaciones. Es posible sólo trabajar con tres particiones, sobre todo si sólo van a acceder al sistema Linux unos pocos usuarios. Estos son algunos consejos importantes:

Los sistemas de archivos importantes que deben estar en particiones independientes son la raiz (/), /var y /tmp desde el punto de vista de la seguridad, o la raiz (/), var y /usr desde el punto de vista administrativo.

En base ha esto se decidió ubicar el directorio /home /usr y /var en particiones diferentes a la de la raíz.

El tamaño de las particiones dependerá del tamaño del disco duro y del uso que se le dará al sistema, la única partición que tendrá un tamaño predefinido será la partición de memoria swap que tendrá un tamaño por lo menos 2 veces a la cantidad de memoria RAM.

De acuerdo a las necesidades y características del equipo con el que se cuenta se propone la siguiente tabla de particiones, tomando en cuenta que la forma en como serán ubicadas estas particiones aumentara le desempeño del servidor web:

- Partición para raiz /.
- Partición Swap.
- Partición para el directorio /usr
- Partición para el directorio /home.
- Partición /var.

La tabla de particiones para el servidor web quedo de la siguiente manera:

- 1era. partición 4000MB para el sistema /.
- 2da. partición 4000MB para /home.
- 3ra. partición 4000MB para /usr.
- 4ta. partición 3800MB para /var
- 5ta. partición 256MB para swap.

La creación de particiones Linux no es algo trivial como en Windows, así es que se hará un pequeño paréntesis en ésta parte del capitulo para mostrar como se crea una partición en Linux.

La herramienta fdisk cuenta con un menú donde se puede observar cada una de las opciones con las que cuenta ésta herramienta, las opciones más importantes son:

- **p** - despliega la tabla de partición actual.
- **m** - despliega el menú de las opciones.
- **d** - elimina una partición.
- **n** - crear una nueva partición.
- **t** - cambia el sistema de identificación de la partición.
- **q** - sale de fdisk sin guardar los cambios.
- **w** - escribe los cambios en el disco y sale de fdisk.

Para crear una nueva particiones se oprime la tecla n fdisk pregunta si se quiere crear una partición primaria o extendida, como la primera partición será para el sistema "/" entonces se oprime la tecla p para una partición primaria.

Se tiene que llevar un orden en las particiones así es que fdisk preguntara por el numero de la partición (1-4), se oprime 1 después enviara un mensaje con el numero del primer cilindro que se encuentra disponible sólo se da un enter para tomar el valor que se indica por omisión, enseguida preguntara por el ultimo cilindro, el cual se puede especificar en megas de la siguiente manera +tamañoM.
Donde tamaño es la cantidad en megas para el tamaño de la partición por ejemplo:

+2000M - partición de 2Gb.
+3500M - partición de 3.5Gb.

Ya que se tiene la primera partición, se puede continuar con las otras, para crear las demás particiones se siguen los mismos pasos.

Cuando se haya terminado de crear las cuatro particiones estas aparecerán como Linux nativa, se debe cambiar el tipo de partición que será la partición swap (**128MB**).

Se oprime la letra t para cambiar el identificador, se indica el numero de la partición (1-4) que se va a modificar y se selecciona el tipo de identificador de una lista oprimiendo la tecla **t**, se elige el numero de identificador, que seria **82** y se da un enter.

Ahora ya se tienen las cuatro particiones se guardan los cambios, con la tecla w y sé continua con la instalación.

```
   Device Boot      Start         End      Blocks   Id  System
/dev/hda2               1         255     2048256   83  Linux native
/dev/hda3             256         272      136552+  83  Linux native
/dev/hda4             273         522     2008125    5  Extended
/dev/hda5             273         400     1028128+  83  Linux native
/dev/hda6             401         515      923706   83  Linux native

Command (m for help): t
Partition number (1-6): 3
Hex code (type L to list codes): 82
Changed system type of partition 3 to 82 (Linux swap)

Command (m for help): p

Disk /dev/hda: 255 heads, 63 sectors, 522 cylinders
Units = cylinders of 16065 * 512 bytes

   Device Boot      Start         End      Blocks   Id  System
/dev/hda2               1         255     2048256   83  Linux native
/dev/hda3             256         272      136552+  82  Linux swap
/dev/hda4             273         522     2008125    5  Extended
/dev/hda5             273         400     1028128+  83  Linux native
/dev/hda6             401         515      923706   83  Linux native

Command (m for help): _
```


Figura 2.5 pantalla de fdisk las particiones han sido creadas.

Para ingresar al programa de instalación se teclea setup y se oprime enter.

```
12  Compaq diagnost 54  OnTrackDMG      a0  IBM Thinkpad hi fa  LANstep
14  Hidden FAT16 <3 55  EZ-Drive        a5  BSD/386             ff  BBT
16  Hidden FAT16    56  Golden Bow
Hex code (type L to list codes): 82
Changed system type of partition 3 to 82 (Linux swap)

Command (m for help): p

Disk /dev/hda: 255 heads, 63 sectors, 522 cylinders
Units = cylinders of 16065 * 512 bytes

   Device Boot      Start         End      Blocks   Id  System
/dev/hda2               1         192    1542208+  83  Linux native
/dev/hda3             193         201      72292+  82  Linux swap

Command (m for help): w
The partition table has been altered!

Calling ioctl() to re-read partition table.
Syncing disks.

WARNING: If you have created or modified any DOS 6.x
partitions, please see the fdisk manual page for additional
information.
# setup_
```

Figura 2.6 Arranque de la instalación.

Enseguida aparecerá una pantalla con un menú con diferentes opciones, las cuales se pueden seleccionar con las teclas de cursor y la tecla enter para ingresar a ellas.

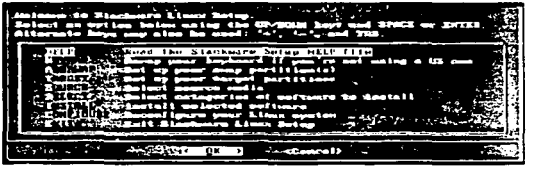


Figura 2.7 Menú de instalación.

Lo primero que se hace es seleccionar la configuración del teclado(KEYMAP).

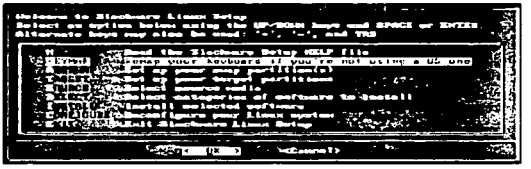


Figura 2.8 Paso 1 teclado.

Se elige el mapa es.map de la lista para configurar el teclado en español y se oprime la tecla enter.

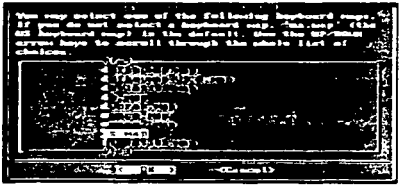


Figura 2.9 Selección teclado español.

Se realiza una pequeña prueba al teclado para comprobar la configuración se escribe cualquier cosa y se elimina para después oprimir la tecla 1 y enter con esto se acepta la configuración.

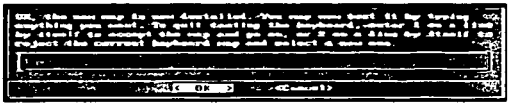


Figura 2.9 Prueba del teclado.

En el siguiente paso el programa de instalación detectara las particiones swap que existan en el disco duro enviara un mensaje en pantalla con las características de ésta, se oprime yes para continuar.

Figura 2.10 Detección de la partición swap.

La partición swap se formatea y activa para su uso inmediato, a diferencia de otras distribuciones Linux Slackware habilita la partición swap primero antes que otras, en caso de que el equipo donde se instala Slackware no tiene suficiente memoria RAM entonces hace uso inmediato de la memoria swap.

Figura 2.11 Formateando partición swap.

Una vez terminado el formato se agrega una linea nueva al archivo de configuración fstab, este archivo indica los dispositivos que serán montados al arranque del sistema se oprime enter para continuar.

Figura 2.12 Swap configurada.

Enseguida el programa de instalación detectara todas las particiones nativas que existan en el disco duro, aparecerá una lista con cada una de las particiones nativas y sus características la primera partición será donde se instale el sistema "/"se selecciona y se da un enter.

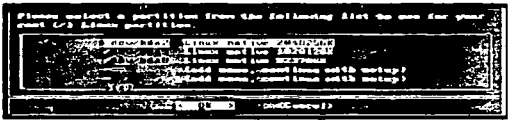


Figura 2.13 Selección de la partición raiz.

En la siguiente pantalla se elige una de tres opciones para dar formato a la partición, se elige la segunda para cerciorase que no existan bloques dañados y se oprime enter para continuar.

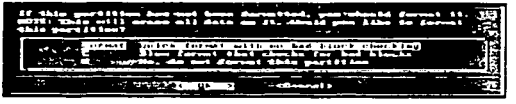


Figura 2.14 Preparación de la partición.

Se necesita seleccionar la densidad del inodo, se elige la primera opción, como lo indica el programa de instalación.

La partición es dividida en pequeños bloques llamados inodos cada inodo tiene un tamaño que puede variar, pero por omisión se utilizara el tamaño de 4096 bytes que es el estándar en Linux, estos inodos contienen las características de cada archivo y directorio como su tamaño, permisos, dueño, etc. Se elige la primera opción y se oprime enter para continuar.

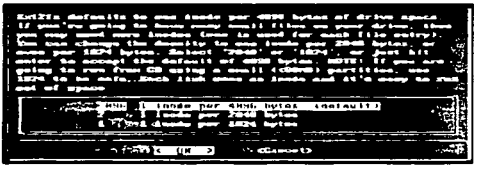


Figura 2.15 Elección del tamaño del inodo.

El programa de instalación dará formato a la primera partición, esto tardara algunos minutos mientras se verifica que no existan bloques malos.

Figura 2.16 Formateando la partición principal.

Después se habilitara la primera partición como partición de arranque, ahora sólo falta agregar las dos particiones restantes para cada una de ellas se siguen los mismos pasos que los anteriores, pero con la diferencia de que se les debe asignar un punto de montaje. Como ya se había mencionado la segunda partición se montara en /home y la tercera en /var.

Figura 2.17 Designando el punto de montaje.

Al final cada partición quedara lista para su uso y se agregaran lineas nuevas al archivo fstab, para que sean montadas al arranque del sistema.

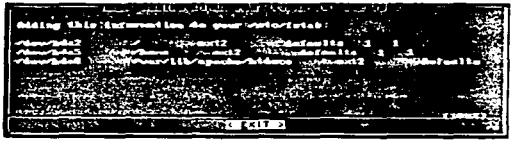


Figura 2.18 Contenido del archivo fstab.

Continuando con la instalación se debe seleccionar el lugar de donde se obtendrán los paquetes, no hay que dar una gran explicación para esto, solamente tomamos la primera opción por obvias razones y se da un enter.

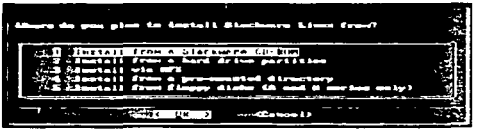


Figura 2.19 Seleccionando el dispositivo desde donde se instalara.

Llego el momento de seleccionar los paquetes necesarios para que el equipo trabaje como servidor de paginas web, si se oprime la tecla espaciadora se puede agregar o quitar paquetes por omisión la mayoría de los paquetes se encuentran seleccionados. De la lista sólo se agregaran los paquetes de la serie A, Ap, D, F, K y N.

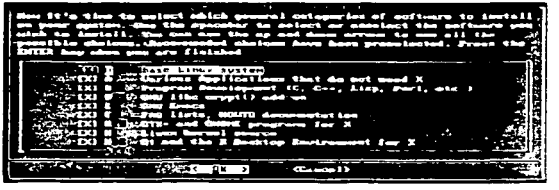


Figura 2.20 Selección de los paquetes.

Una vez terminada la selección de paquetes se elige el tipo de instalación, se toma la primera Slackware y se oprime enter.

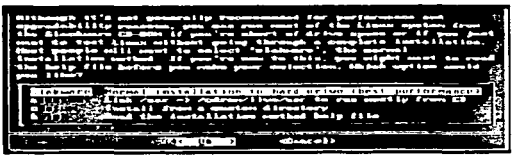


Figura 2.21 Elección del tipo de instalación.

En la siguiente ventana se elige la primera opción, para que todo se instale en forma completa, sin preguntas y en forma rápida.

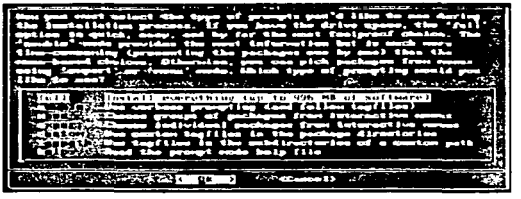


Figura 2.22 Elección del modo de interacción.

Ahora sólo falta espera que los paquetes seleccionados se instalen para continuar con la ultima parte de la instalación.

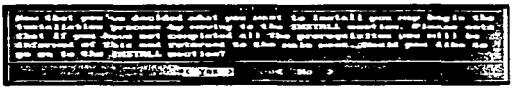


Figura 2.23 Terminada la instalación se procede a la configuración.

Cuando comienza la instalación de los paquetes veremos pasar ventanas con algunos mensajes que contienen las caracteristicas de los paquetes, una pequeña descripción y tamaño de cada paquete en forma individual.

Cuando todos los paquetes seleccionados hayan terminado de instalarse aparecerá una venta donde se puede crear un disco de arranque, esto no es necesario y en pocas ocasiones se llega a emplear, ya que se puede utilizar el CD-ROM para arrancar el sistema y darle mantenimiento, se puede saltar ésta parte eligiendo la ultima opción.

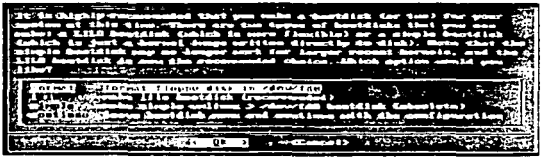


Figura 2.24 Crear un disco de arranque.

El equipo no contiene ningún tipo de módem ya que la conexión será a través de un enlace dedicado, se elige la opción no módem y se oprime enter.

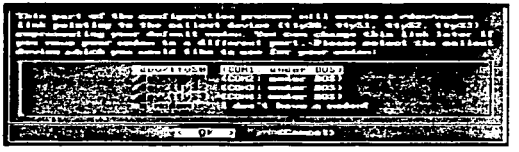


Figura 2.25 Selección de un MODEM.

En ésta pantalla se puede seleccionar un tipo de letra diferente con el que se ésta trabajando en la instalación de ésta forma si se selecciona un tipo diferente este tipo de letra aparecerá cuando finalice la instalación y el sistema arranque nuevamente el equipo este paso se puede omitir solamente dando enter.

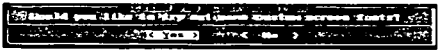


Figura 2.26 Selección del tipo de fuente.

La instalación de lilo es uno de los pasos de la instalación más importantes se debe elegir de entre dos opciones simple y experto, se elige la primera opción y se teclea ok.

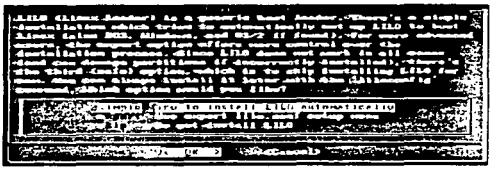


Figura 2.27 Modo de instalación del menú de arranque LILO.

Lilo se alojará en el MBR (Master Boot Record) que es el primer sector donde lee la bios para saber que SO debe arrancar o en el inicio de la partición raíz, se oprime enter para continuar.

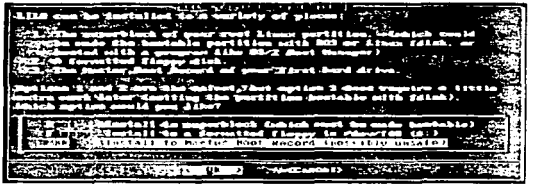


Figura 2.28 Ubicación del programa LILO.

La siguiente ventana es la configuración de los dispositivos de red este paso se puede omitir, para configurar la red mas adelante se selecciona no para continuar.

Figura 2.29 Configuración de la red.

El siguiente paso es seleccionar el tipo de ratón instalado en el equipo que es un ratón serial, el programa de instalación lo detecta automáticamente se selecciona ok para continuar.

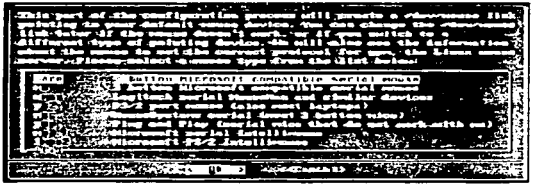


Figura 2.30 Selección del tipo de ratón.

Para finalizar con la configuración del ratón se elige el puerto COM correcto donde ésta instalado fisicamente se elige la primera opción y se teclea ok para continuar.

Figura 2.31 Elección del puerto del ratón.

El GPM es un programa en que permite poder trabajar en modo texto con el ratón se puede cortar, copiar y pegar con el es muy útil cuando se editan archivos de configuración se oprime ok para instalarlo y se continua con la instalación.

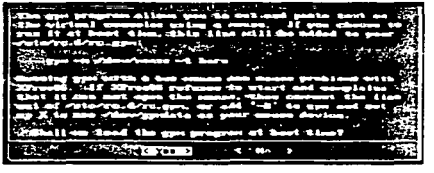


Figura 2.32 Instalación de GPM.

Se puede habilitar a sendmail como servidor de correo electrónico, elegimos la primera opción ya que se cuenta con el servicio de DNS proporcionado por la UNAM, de ésta manera una vez instalado el servidor se puede hacer uso de este servicio se selecciona ok para continuar.

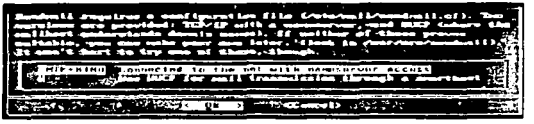


Figura 2.33 Configuración de sendmail.

No se utilizara el servicio UTC así es que se elige la primera opción del menú para utilizar solamente el horario proporcionado por el sistema se oprime ok y se continua con la instalación.

Figura 2.34 Modo de sincronización del reloj del sistema.

Para seleccionar la zona horario correcta para nuestro equipo se debe seleccionar uno de la lista que aparece en ésta venta, se elige America/Mexico_City. y se oprime ok para continuar.

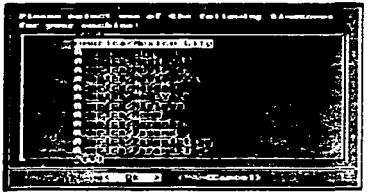


Figura 2.35 Configuración de la zona horario.

Para finalizar con la instalación se debe proporcionar un password al superusuario que es root, se debe elegir un password lo bastante complejo para que no pueda ser descifrado fácilmente por cualquier persona extraña que quiera ingresar al sistema, debe tener entre 6 y 8 caracteres, combinar letras mayúsculas, minúsculas, signos de puntuación caracteres especiales, etc. Una vez ingresado el password para root se teclea ok para finalizar con la instalación.

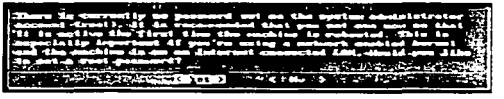


Figura 2.36 Definición del password de root.

El programa de instalación enviara un mensaje en pantalla para indicar que la instalación ha terminado y que se puede reiniciar el equipo para poder ingresar al sistema se teclea ok para continuar.

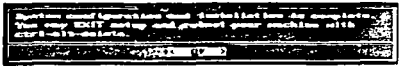


Figura 2.37 Configuración completa y listo para reiniciar.

Inmediatamente después aparecerá nuevamente la pantalla que aparece la principio de la instalación se selecciona la opción exit para salir del programa y se oprime ok.

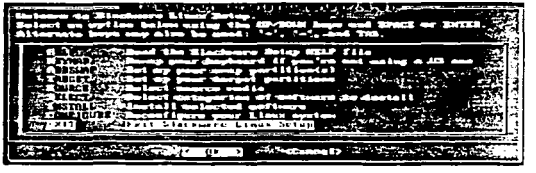


Figura 2.38 Menú inicial.

Por ultimo se necesita reiniciar el sistema para poder ingresar al sistema y saber si funciona correctamente se escribe en la linea de comandos la siguiente instrucción:
#reboot

- Con esto el sistema rearrancara nuevamente, se debe quitar el disco de instalación de Slackware de la unidad y con esto se da por finalizada la instalación.

2.5 Configuración del sistema.

Ahora que ya se tiene instalado Linux es necesario realizar algunas modificaciones al sistema antes de que este listo para funcionar como servidor de paginas web, estas tareas son:

1. **Creación de cuentas de usuario**
2. **Configuración de los dispositivos de red.**

Para realizar cualquier modificación al sistema es necesario contar con permisos de root.

2.5.1 Cuentas de usuario.

Crear una cuenta de usuario en Linux es bastante fácil se puede hacer de tres formas distintas utilizando una herramienta gráfica, editando archivos de configuración o a través de la línea de comandos. Esta ultima es la forma más fácil de crear cuentas se utiliza la herramienta adduser para este propósito.

Esta herramienta hace todo el trabajo necesario para crear una cuenta lo único que debe saber el administrador es el login y contraseña que se asignara a cada usuario, la forma de arrancar adduser es la siguiente. Se Ingresa al sistema con la cuenta root y en la línea de comandos se escribe adduser y se oprime enter.

Trabajar con adduser es bastante fácil y no es necesario explicar a detalle su funcionamiento para eso existen libros completos de administración que detallan el uso de ésta herramienta, en el servidor web existen varias cuentas:

1. **La cuenta personal del administrador**
2. **El encargado de la página web**
3. **La del jefe de departamento.**

Una vez creadas todas las cuentas que habrá en el sistema, es momento de continuar con el siguiente paso configurar los dispositivos de red.

2.5.2 Configuración de los dispositivos de red.

Para configurar los dispositivos de red que están instalados en el servidor es necesario utilizar la herramienta netconfig sólo se debe saber algunos datos como son:

- **Nombre del host**
- **Nombre de dominio**
- **Dirección IP**
- **Puerta de enlace.**
- **Servidor de nombre de dominio**

Para ejecutar netconfig solamente tecleamos el nombre de la herramienta en la línea de comandos y se oprime enter.

Figura 2.39 Pantalla de inicio de netconfig.

La herramienta netconfig preguntara cada uno de los parámetros ya antes mencionados sólo se debe escribir los datos correctos y oprimir enter para continuar con la configuración.

Al final netconfig realizara un test para detectar la tarjeta de red y elegir el controlador correcto.

Figura 2.40 Detección de la tarjeta de red.

Una vez detectada la tarjeta ésta queda configurada y lista, después de haber realizado estas modificaciones el servidor queda con una configuración mínima para funcionar y comenzar a trabajar en red.

CAPÍTULO 3
SERVICIOS

3. Servicios

Al terminar la instalación de Linux, después de crear las cuentas de usuario y configurar la tarjeta de red el servidor quedo listo para trabajar en red, en este momento el servidor puede ya proporcionar diferentes servicios, una de las caracteristicas de Slackware es la de habilitar todos los servicios de red con los que cuenta y dejar para el administrador la tarea de seleccionar aquellos que le sirvan para sus tareas diarias y deshabilitar aquellos que no le sean necesarios.

En ocasiones por desconocimiento del administrador del sistema no sabe que servicios tiene ejecutando en su sistema y mucho menos sabe como eliminar estos servicios, esta parte se tratara afondo en el tema de seguridad. En este capitulo es analizaran las caracteristicas de los servicios que están corriendo y de esta manera tomar una decisión para seleccionar aquellos que sirvan para la instalación del servidor web, además que puedan ser útiles en la administración del servidor.

Se necesita para administrar el servidor contar con servicios que permitan conexiones remotas, correo electrónico y subir archivos además de un software lo bastante bueno como para instalar un servidor de paginas web, una caracteristica importante será que no sea un problema poder conseguir una versión de este software ni que sea tan complicada su instalación y configuración.

Por fortuna Linux Slackware cuenta con todos los servicios necesarios para desempeñar estos trabajos, se encuentran instalados y configurados solo resta elegir algunos e eliminar a los demás.

De acuerdo con estas necesidades se han seleccionado cuatro servicios que cumplen con estas caracteristicas, cada uno se encargara de una tarea especifica. A continuación se da una pequeña explicación de cada uno de ellos.

3.1 Telnet.

Es un protocolo que permite conexiones remotas desde una maquina cliente hacia un servidor por medio de TCP, este programa existe en todas las distribuciones Linux y UNIX lo cual lo hace un estándar.
La conexión se realiza por medio de un programa cliente que se comunica con el host remoto (servidor), telnet utiliza el puerto 23 para entablar la comunicación con el host, todas las peticiones de clientes telnet se dirigen a ese puerto, telnet necesita de un login y una contraseña para poder realizar la conexión en caso contrario el host rechazara la conexión.

Una vez que se realiza la conexión el usuario puede trabajar en su home de trabajo que existe en el host remoto donde puede crear, borrar, almacenar archivos y directorios siempre que le pertenezcan, si un usuario desea entrar a un home de trabajo que no es el suyo el sistema no lo permitirá.

La mayor ventaja de telnet es que existe un cliente para cada SO esto hace que su uso sea extenso.

Existen otros programas que se pueden utilizara para este propósito como rlogin pero su uso se limita a ambientes UNIX lo cual hace que usuarios windows queden relegados.

De acuerdo con la anterior telnet será el programa que proporcione la conexión remota hacia el servidor.

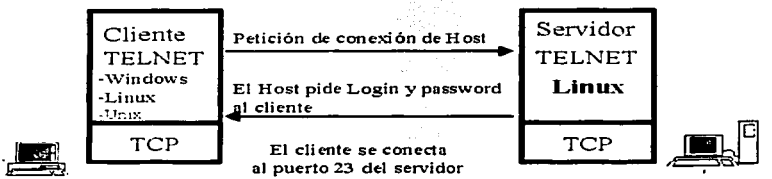


Figura 3.1 Conexión TELNET.

3.2 FTP.

Junto con telnet ftp es el protocolo que se utiliza desde hace mucho tiempo para la transferencia de archivos, permite realizar una conexión remota con un host por medio de TCP pero a diferencia de telnet ftp fue diseñado para poder subir y descargar archivos desde un host.

Hay dos formas de poder realizar una conexión remota con ftp que son la conexión publica y privada, la primera permite que cualquier cliente tenga acceso al sistema pero solo a un directorio especifico para este propósito, el usuario que ingresa con esta cuenta solo podrá trabajar en este directorio y no podrá acceder a mas recursos, la forma privada es similar a una conexión telnet se necesita un login y una contraseña para ingresar una vez obtenido el acceso el usuario será alojado en su home de trabajo donde podrá subir y descargar los archivos que el desee.

El protocolo ftp utiliza el puerto 21 para realizar conexiones a un host remoto, soporta diferentes tipo de sistemas de archivos, existe un cliente para cada SO, y es un estándar para la transferencia de archivos por estas razones se elige a ftp como el protocolo para la transferencia de archivos en el servidor web.

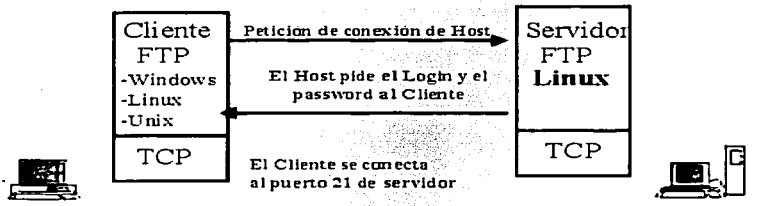


Figura 3.2 Conexión FTP.

3.3 Correo electrónico.

Uno de los servicios más importantes para cualquier servidor en Internet es el servicio de correo electrónico, hoy en día cualquier persona tiene una cuenta de correo electrónico, para este fin se utilizara a sendmail que se cataloga como uno de los mejores programas de correo electrónico, este software viene incluido en la distribución de Linux Slackware y se encuentra disponible en la mayoría de las distribuciones Linux. Además no hay que hacer mucho para hacerlo funcionar ya que durante la instalación se configuro y se encuentra listo para comenzar a trabajar. Basta con tener una cuenta valida en el servidor y una conexión a Internet para comenzar a enviar correo electrónico.

Sendmail incluye un archivo de configuración que contiene habilitadas algunas opciones mínimas para comenzar a trabajar con él, además sendmail soporta los protocolos SMTP y POP con esto se asegura la compatibilidad con cualquier cliente de correo electrónico sin importar el SO.

Por estas razones sendmail es el software que se utilizara para habilitar un servidor de correo electrónico en el servidor web y así poder también proporcionar este servicio a los usuarios.

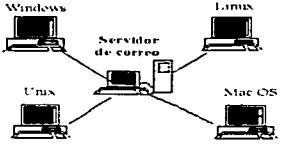


Figura 3.3 Correo electrónico.

3.4 HTTP.

El protocolo http se utiliza para poder realizar conexiones remotas con un host que proporcioné un servicio de paginas web, utilizando un cliente (que es un navegador), proporcionando un identificador URL de la página que se desea obtener, el cliente hace una petición al servidor al puerto 80 una vez que se logra la conexión el servidor responde al cliente enviando la pagina que se busca, o con un mensaje si no se encuentra la página.

Cuando se logra la conexión con un servidor web el cliente envia instrucciones que interpreta el servidor web, como si se tratara de una sesión telnet.

El cliente para poder acceder a los diferentes recursos de un servidor web se le conoce con el nombre de navegador, existen diferentes navegadores en modo texto y gráficos como son lynx, netscape, conqueror, mosaic y explorer.

En todas las distribuciones Linux se incluye un software especial para poder instalar un servidor de páginas web en Linux, desde hace mucho tiempo este software ha demostrado ser el mejor de todos en este ramo, desarrollado por el proyecto Apache este software esta por encima del IIS de Microsoft, apache que es el nombre como se le conoce a este software permite poder montar un servidor web en Linux en muy poco tiempo, su archivo de configuración es un archivo con algunas directivas fáciles de modificar que permite personalizar a este software tanto como se desee, inclusive si no se tiene una gran experiencia en el manejo de archivos de configuración se puede dejar este archivo tal y como se incluye dentro de la distribución y apache funcionara sin problemas.

Una caracteristica importante de apache es su potabilidad existen versiones para diferentes SO incluyendo a Windows 98, además se distribuye en forma gratuita bajo los términos de la licencia GNU y viene incluido en todos las diferentes distribuciones de Linux lo cual lo hace más atractivo.

Por estas razones apache es el software que se utilizara para instalar el servidor web.

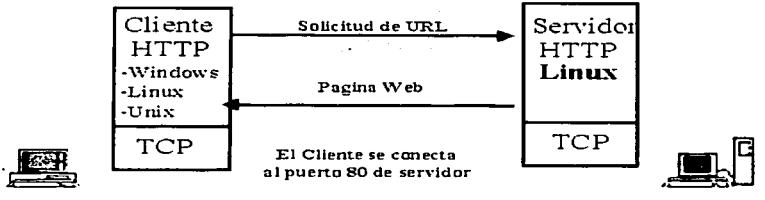


Figura 3.4 Conexión HTTP.

3.5 Servicios básicos.

Los servicios antes mencionados telnet, ftp, correo electrónico, http deberían ser los únicos servicios para cualquier servidor web, todo dependerá de las características de cada servidor y su función.

Pero de acuerdo con la experiencia obtenida se propone que estos son los servicios mínimos y necesarios que debería estar proporcionando un servidor web sea cual sea la distribución, aunque algunos de estos servicios (telnet y ftp) no son los mejores en cuestión de seguridad si son algunos de ellos un estándar en la mayoría de los SO, también son compatibles con diferentes tipos de software (outlook, eudora, Netscape, explorer) y cualquier persona con conocimientos mínimos de cómo manejar una computadora los puede utilizar.

Algunos otros servicios como NFS, Samba y lpd no son necesarios en el servidor es por eso que se deshabilitaran pero se recomienda saber que es lo que hace cada servicio antes de deshabilitarlo y estar seguro que es un servicio que no se utiliza y que se puede dar de baja sin que afecte en el funcionamiento de los demás servicios, no esta de mas conocer algunas de sus características.

Enseguida una pequeña semblanza de algunos de estos servicios para que las personas que nunca los han manejado ni escuchado los conozcan, ya que es importante para cualquier administrador de sistemas conocer a fondo el SO con el que esta trabajando.

3.6 NFS.

El protocolo NFS o Network File System permite que una maquina cliente pueda importar un directorio completo de un host remoto (servidor), el host remoto permitirá exportar algún directorio de su sistema de archivos siempre y cuando la máquina cliente este definida dentro del archivo de acceso del host.

El NFS se utiliza en casos de maquinas con pocos recursos se instala una versión pequeña de Linux en ella, los directorios de trabajo de cada usuario y las aplicaciones se almacenan y ejecutan en el servidor, en este caso las computadoras cliente funcionaran como terminales tontas.

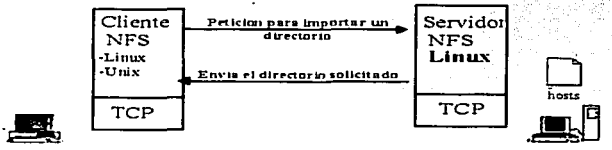


Figura 3.5 NFS.

3.7 Samba.

El protocolo samba fue inventado para hacer posible que máquinas con UNIX puedan compartir recursos con máquinas Windows de la misma manera en como lo hacen estas, samba es nativo de los SO Windows pero se escribió una versión para UNIX que también se encuentra en Linux, Slackware cuenta con una versión de samba instalada y corriendo solo hace falta modificar algunos parámetros de su archivo de configuración para compartir recursos, con samba se puede compartir desde directorios completos hasta impresoras instaladas en Linux.

Con samba también es posible acceder a recursos que están compartidos en máquinas Windows.

3.8 LPD.

Este es el servidor de impresoras compatibles con UNIX (postscript) se encarga de enviar todos los trabajos de impresión a las impresoras que estén instaladas en el servidor.

Estos son solo algunos de los servicios que Linux Slackware esta ejecutando, existen muchos mas pero aquí solo mencionamos los mas importantes, para mayor referencia acerca de los servicios que proporciona Linux se puede consultar el archivo /etc/services que contiene una lista de los servicios que Linux puede proporcionar, además de otras características como el numero de puerto y tipo de protocolo que utilizan.

CAPÍTULO 4
ADMINISTRACION DE UN SERVIDOR WEB

En este capítulo se hablara del servidor de WWW apache, origen, características, diferentes plataformas donde puede correr y la forma de cómo obtenerlo.

La distribución de Linux Slackware incluye una versión del servidor apache ya instalada y corriendo, antes de entrar la configuración de este software, es necesario conocer un poco mas a fondo algunas de sus características y origen.

4.Origen del servidor apache.

Antes de que el servidor apache fuera lo que es hoy, un grupo del Centro Nacional de Actividades de Supercomputación, desarrollo el servidor NCSA de HTTPd y el navegador gráfico Mosaic, este servidor fue el mas utilizado en los primeros años de su desarrollo, pero en el año de 1994 el principal desarrollador del servidor NCSA dejo el proyecto.

En el año de 1995 se crea un nuevo grupo de trabajo para continuar con el proyecto pero ahora con el nombre de apache, diferentes personas de todo el mundo se involucraron en este proyecto haciendo que apache se desarrollara mucho más rápido, en el mes de abril de 1995 se lanza la primera versión de apache.

El software apache ha recibido algunos premios demostrando la calidad de este software estos premios se pueden observar en la pagina principal del proyecto apache (www.apache.org).

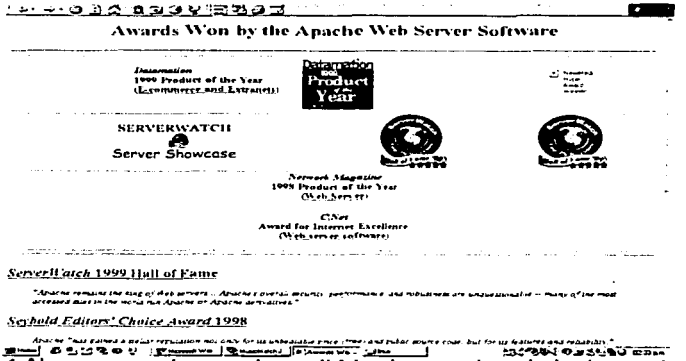


Figura 4.1 Algunos premios que ha recibido el grupo de trabajo de apache por su labor.

De acuerdo con estadísticas proporcionadas por netcraft, que es una empresa que realiza encuestas desde 1995 con carácter mensual, demuestran que apache se emplea mas que el resto de los servidores web, lo cual se puede observar en la siguiente gráfica.

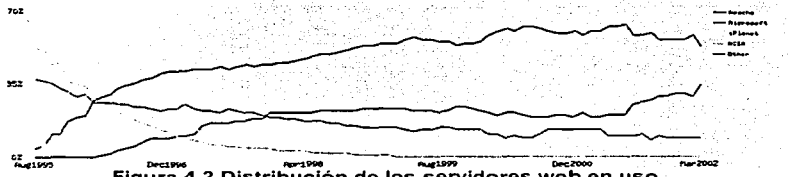


Figura 4.2 Distribución de los servidores web en uso.

A continuación se muestran algunas estadísticas obtenidas de la página de netcraft (www.netcraft.com/survey), acerca de los cuatro servidores que más se usan en Internet.

Top Developers

Developer	February 2002	Percent	March 2002	Percent	Change
Apache	22462777	58.43	20492088	53.76	-4.67
Microsoft	11198727	29.13	12968860	34.02	4.89
Iplanet	1123701	2.92	889857	2.33	-0.59
Zeus	837968	2.18	855103	2.24	0.06

Active Sites

Developer	February 2002	Percent	March 2002	Percent	Change
Apache	10147402	65.18	9522954	64.37	-0.81
Microsoft	4069193	26.14	3966743	26.81	0.67
Iplanet	283112	1.82	265826	1.80	-0.02
Zeus	177225	1.14	170023	1.15	0.01

Figuras 4.3 Servidores que mas se utilizan en Internet.

La siguiente figura 4.4 muestra la distribución de los mejores servidores que se están utilizando y del número de sitios web que utilizan estos servidores

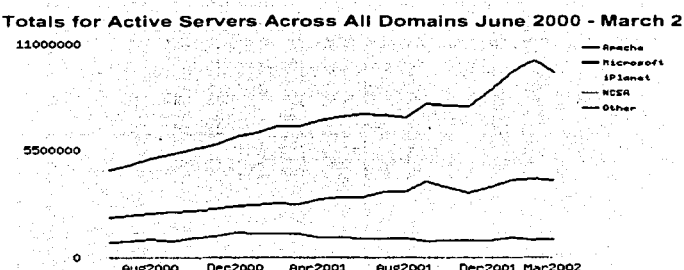


Figura 4.4 Distribución total de todos de los servidores web en uso.

4.1 Distribución de apache.

Apache esta disponible en diferentes presentaciones para UNÍX y LINUX, en forma binaria (precompilada) y con su código fuente, también se pueden encontrar algunas versiones para Windows, para alegría de las personas que trabajan en este sistema, cuenta con un asistente para su instalación que hace relativamente fácil la instalación de apache en Windows.

Para los usuario UNIX y para sus clones como LINUX es más complejo este proceso que estar solo dando clic, pero se cuenta con un manual de instalación que se puede consultar directamente en la pagina principal de apache (www.apache.org) o dentro de la copia del software que se descarga de este lugar, esta son las mejores referencias que se pueden consultar para la instalación de apache.

En la figura 4.3 se puede observar la página web donde puede elegirse la distribución y versión de apache que sea la mas indicada para cubrir las necesidades de cada usuario, considerando como se menciono la plataforma y SO con que se cuenta, así como una distribución ya compilada o la versión para compilar en el propio equipo.

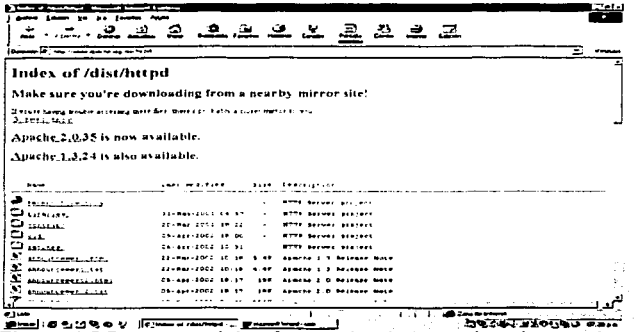


Figura 4.5 Descarga del software desde la pagina de apache.

Se recomienda que cuando se descargue cualquier versión de apache. Esta sea la mas actual, existe una liga en la pagina de apache donde se pueden consultar los diferentes problemas que se han encontrado en cada versión, hay que consultar esta liga para asegurarse que la versión que se descargue no contiene errores.

4.1.1. Requerimientos mínimos.

Los requerimientos mínimos para instalar apache son como mínimo 12MB de espacio temporal en disco duro para la instalación, apache ocupara ya instalado alrededor de 3MB, además del espacio que se utilice para instalar la página web y un compilador C ANSI o GCC.

La mayoría de los Linux cumplen con estos requerimientos así es que no debería haber ningún problema para su instalación.

En caso de que no se cuente con un compilador, es posible instalar una versión binaria acorde al UNÍX o LINUX que se este empleando, no se recomienda utilizar una distribución binaria ya que esta distribución esta compilada con ciertas características que pueden ser suficientes o no para nuestras necesidades es mejor instalar una distribución fuente la cual podemos personalizar de acuerdo con nuestras necesidades y características del sistema.

4.2 Configuración.

Linux Slackware incluye una versión de apache instalada, compilada y trabajando, solo hace falta editar algunos archivos de configuración para personalizarlo y dejarlo lista para trabajar con él.

Para saber si verdaderamente apache esta corriendo en el servidor solamente basta con abrir un navegador sea cual sea y teclear en el cuadro de direcciones la dirección IP del propio equipo o el nombre del equipo asignado al servidor, el navegador desplegara una pagina de bienvenida al servidor de apache, como la siguiente.

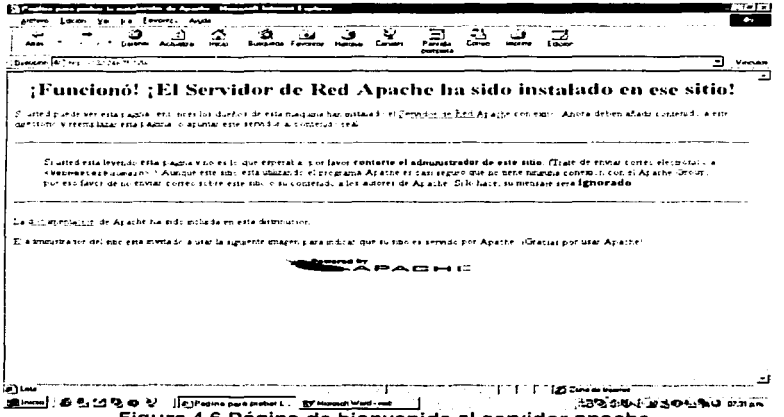


Figura 4.6 Página de bienvenida al servidor apache.

Si se logra ver esta pantalla, entonces ya se tiene listo el servidor web para trabajar.

4.3 Administración del servidor web.

La administración del servidor web resulta una tarea sencilla una vez que se sabe donde se encuentran ubicados los directorios y archivos que componen el software de apache, la versión de Slackware 7.1 ubica estos directorios y archivos dentro del directorio que se encuentra en /var/lib/apache, enseguida se mencionarán algunas caracteristicas de estos directorios y su importancia para el servidor web.

Enseguida se muestra el contenido del directorio apache.

```
usuario@hermes:/var/lib/apache$ ls -l
total 248
-rw-r--r--  1 root   root    12957 Mar 31  1999 ABOUT_APACHE
-rw-r--r--  1 root   root     2922 Feb 23  2000 Announcement
-rw-r--r--  1 root   root    27964 Dec 21  1999 INSTALL
-rw-r--r--  1 root   root    35773 Aug 20  1999 KEYS
-rw-r--r--  1 root   root     2848 Jan  1  1999 LICENSE
-rw-r--r--  1 root   root    26758 Jan 11  2000 Makefile.tmpl
-rw-r--r--  1 root   root     2046 Apr  1  1998 README
-rw-r--r--  1 root   root     3132 Mar 19  1999 README.NT
-rw-r--r--  1 root   root    11176 Dec 20  1999 README.configure
-rw-r--r--  1 root   root      331 Sep 21  1998 WARNING-NT.TXT
drwxr-xr-x  2 root   root     4096 Jun 14  2000 bin/
drwxr-xr-x  2 root   root     4096 Jun 14  2000 cgi-bin/
drwxr-xr-x  2 root   root     4096 Feb 15 20:15 conf/
-rw-r--r--  1 root   root     6519 Jun 14  2000 config.layout
-rwxr-xr-x  1 root   root    56062 Feb  5  2000 configure*
drwxr-xr-x 14 root   root     4096 Feb  1 20:40 htdocs/
drwxr-xr-x  3 root   root     4096 Jun 14  2000 icons/
drwxr-xr-x  3 root   root     4096 Jun 14  2000 include/
drwxr-xr-x  2 root   root     4096 Feb 15 20:15 libexec/
drwxr-xr-x  4 root   root     4096 Jun 14  2000 man/
drwxr-xr-x  2 root   root     4096 Jun 14  2000 sbin/
drwxr-xr-x 11 root   root     4096 Jun 14  2000 src/
```

Figura 4.7 Contenido del directorio /var/lib/apache

Los directorios importantes son:

Directorio	Descripción
/bin	Este directorio almacena algunos archivos binarios.
/cgi-bin	Este directorio almacena los programas cgis.
/conf	Este directorio almacena los archivos de configuración, como el httpd.conf.
/htdocs	Este directorio almacena los archivos con extensión html, como el index.html.
/sbin	Este directorio almacena los archivos binarios que sirven para la administración del servidor web.

Tabla 4.1 directorios principales de apache.

Una vez que se sabe cual es el contenido de estos directorios y su propósito dentro del servidor web se puede comenzar a realizar modificaciones al servidor.

4.3.1 Archivos de configuración.

En sus inicios el servidor NCSA distribuía la administración del servidor web en tres archivos access.conf, httpd.conf y srm.conf, esto funcionaba en versiones antiguas de apache las versiones modernas de apache centralizan la administración del servidor en un solo archivo de configuración el archivo httpd.conf, este archivo de configuración esta dividido en tres secciones. Global Environment, Main server configuration y virtual host.

Para darse una idea del contenido del archivo httpd.conf, a continuación se muestra un extracto del contenido de este archivo.

```
# Based upon the NCSA server configuration files originally by Rob #McCool.
#
# This is the main Apache server configuration file.  It contains the
# configuration directives that give the server its instructions.
# See <URL:http://www.apache.org/docs/> for detailed information about
# the directives.
#
# Do NOT simply read the instructions in here without understanding
# what they do.  They're here only as hints or reminders.  If you are unsure
# consult the online docs. You have been warned.
#
# After this file is processed, the server will look for and process
# C:/Archivos de programa/Apache Group/Apache/conf/srm.conf and then C:/Archivos
de programa/Apache Group/Apache/conf/access.conf
# unless you have overridden these with ResourceConfig and/or
# AccessConfig directives here.
#
# The configuration directives are grouped into three basic sections:
#  1. Directives that control the operation of the Apache server process as a
#     whole (the 'global environment').
#  2. Directives that define the parameters of the 'main' or 'default' server,
#     which responds to requests that aren't handled by a virtual host.
#     These directives also provide default values for the settings
#     of all virtual hosts.
#  3. Settings for virtual hosts, which allow Web requests to be sent to
#     different IP addresses or hostnames and have them handled by the
#     same Apache server process.
#
# Configuration and logfile names: If the filenames you specify for many
# of the server's control files begin with "/" (or "drive:/" for Win32), the
# server will use that explicit path.  If the filenames do *not* begin
# with "/", the value of ServerRoot is prepended -- so "logs/foo.log"
# with ServerRoot set to "/usr/local/apache" will be interpreted by the
# server as "/usr/local/apache/logs/foo.log".
#
```

```
# NOTE: Where filenames are specified, you must use forward slashes
# instead of backslashes (e.g., "c:/apache" instead of "c:\apache").
# If a drive letter is omitted, the drive on which Apache.exe is located
# will be used by default.  It is recommended that you always supply
# an explicit drive letter in absolute paths, however, to avoid
# confusion.
#

### Section 1: Global Environment
#
# The directives in this section affect the overall operation of Apache,
# such as the number of concurrent requests it can handle or where it
# can find its configuration files.
#

#
# ServerType is either inetd, or standalone.  Inetd mode is only supported on
# Unix platforms.
#
ServerType standalone

#
# ServerRoot: The top of the directory tree under which the server's
# configuration, error, and log files are kept.
#
ServerRoot "/var/lib/Apache"

#
# PidFile: The file in which the server should record its process
# identification number when it starts.
#
PidFile logs/httpd.pid

#
# ScoreBoardFile: File used to store internal server process information.
# Not all architectures require this.  But if yours does (you'll know because
# this file will be  created when you run Apache) then you *must* ensure that
# no two invocations of Apache share the same scoreboard file.
#
ScoreBoardFile logs/apache_runtime_status

#
# In the standard configuration, the server will process httpd.conf (this
# file, specified by the -f command line option), srm.conf, and access.conf
# in that order.  The latter two files are now distributed empty, as it is
# recommended that all directives be kept in a single file for simplicity.
# The commented-out values below are the built-in defaults.  You can have the
```

```
# server ignore these files altogether by using "/dev/null" (for Unix) or
# "nul" (for Win32) for the arguments to the directives.
#
#ResourceConfig conf/srm.conf
#AccessConfig conf/access.conf

#
# Timeout: The number of seconds before receives and sends time out.
#
Timeout 300

#
# KeepAlive: Whether or not to allow persistent connections (more than
# one request per connection). Set to "Off" to deactivate.
#
KeepAlive On

#
# MaxKeepAliveRequests: The maximum number of requests to allow


### Section 2: 'Main' server configuration
#
# The directives in this section set up the values used by the 'main'
# server, which responds to any requests that aren't handled by a
# <VirtualHost> definition.  These values also provide defaults for
# any <VirtualHost> containers you may define later in the file.
#
# All of these directives may appear inside <VirtualHost> containers,
# in which case these default settings will be overridden for the
# virtual host being defined.
#

#
# Port: The port to which the standalone server listens.  Certain firewall
# products must be configured before Apache can listen to a specific port.
# Other running httpd servers will also interfere with this port.  Disable
# all firewall, security, and other services if you encounter problems.
# To help diagnose problems use the Windows NT command NETSTAT -a
#
Port 80

#
# ServerAdmin: Your address, where problems with the server should be
# e-mailed.  This address appears on some server-generated pages, such
# as error documents.
#
```

```
ServerAdmin webmaster@www.midominio.com

#
# ServerName allows you to set a host name which is sent back to clients for
# your server if it's different than the one the program would get (i.e., use
# "www" instead of the host's real name).
#
# Note: You cannot just invent host names and hope they work. The name you
# define here must be a valid DNS name for your host. If you don't understand
# this, ask your network administrator.
# If your host doesn't have a registered DNS name, enter its IP address here.
# You will have to access it by its address (e.g., http://123.45.67.89/)
# anyway, and this will make redirections work in a sensible way.
#
# 127.0.0.1 is the TCP/IP local loop-back address, often named localhost. Your
# machine always knows itself by this address. If you use Apache strictly for
# local testing and development, you may use 127.0.0.1 as the server name.
#
ServerName www.midominio.com
#
# DocumentRoot: The directory out of which you will serve your
# documents. By default, all requests are taken from this directory, but
# symbolic links and aliases may be used to point to other locations.
#
DocumentRoot "/var/lib/apache/htdocs"

#
# Each directory to which Apache has access, can be configured with respect
# to which services and features are allowed and/or disabled in that
# directory (and its subdirectories).
#
# First, we configure the "default" to be a very restrictive set of
# permissions.
#
<Directory />
    Options FollowSymLinks
    AllowOverride None
</Directory>

#
# Note that from this point forward you must specifically allow
# particular features to be enabled - so if something's not working as
# you might expect, make sure that you have specifically enabled it
# below.
#

#
```

```
# This should be changed to whatever you set DocumentRoot to.
#

# This may also be "None", "All", or any combination of "Indexes",
# "Includes", "FollowSymLinks", "ExecCGI", or "MultiViews".
#
# Note that "MultiViews" must be named *explicitly* --- "Options All"
# doesn't give it to you.
#
    Options Indexes FollowSymLinks MultiViews

#
# This controls which options the .htaccess files in directories can
# override. Can also be "All", or any combination of "Options", "FileInfo",
# "AuthConfig", and "Limit"
#
    AllowOverride None

#
# Controls who can get stuff from this server.
#
    Order allow,deny
    Allow from all
</Directory>


### Section 3: Virtual Hosts
#
# VirtualHost: If you want to maintain multiple domains/hostnames on your
# machine you can setup VirtualHost containers for them. Most configurations
# use only name-based virtual hosts so the server doesn't need to worry about
# IP addresses. This is indicated by the asterisks in the directives below.
#
# Please see the documentation at <URL:http://www.apache.org/docs/vhosts/>
# for further details before you try to setup virtual hosts.
#
# You may use the command line option '-S' to verify your virtual host
# configuration.

#
# Use name-based virtual hosting.
#
#NameVirtualHost *

#
# VirtualHost example:
# Almost any Apache directive may go into a VirtualHost container.
```

```
# The first VirtualHost section is used for requests without a known
# server name.
#
#<VirtualHost *>
#    ServerAdmin webmaster@dummy-host.example.com
#    DocumentRoot /www/doc/dummy-host.example.com
#    ServerName dummy-host.example.com
#    ErrorLog logs/dummy-host.example.com-error_log
#    CustomLog logs/dummy-host.example.com-access_log common
#</VirtualHost>
```

4.8 Extracto del archivo httpd.conf

El archivo de configuración httpd.conf tiene una sintaxis muy simple todas las líneas que tengan un símbolo de este tipo # serán omitidas por apache, ya que son consideradas como comentarios, todas aquellas que no contengan este símbolo serán tomadas en cuenta por apache para su configuración.

Para realizar modificaciones dentro de este archivo es necesarios utilizar un editor de texto, puede ser uno muy sofisticado como Aviword o uno muy simple como pico.

Las directivas de configuración para el servidor apache están definidas dentro de tres secciones básicas, que se describen a continuación.

1.-Sección Global Environment controla la operación de los procesos del servidor apache.

2.-Sección Main server configuration define como el servidor apache responderá a las peticiones de los clientes web, algunos de estas directivas se encuentran comentadas y cuentan con valores definidos por los desarrolladores de apache.

3.-Sección Virtual hosts permite que el servidor apache responda por una IP o nombre de dominio diferente, utilizando un solo servidor web.

Cada directiva esta definida por un valor que puede ser modificado, acorde con los requerimientos para cada servidor y del administrador.

4.3.2 Funcionamiento de apache.

El funcionamiento de apache se encuentra definido en la primera sección del archivo de configuración httpd.conf, esta sección es bastante extensa y no es necesario conocer todas las directivas que aquí se definen, solo se mencionara las más importantes y su significado.

El encabezado de la seccion es Section 1: **Global Environment**. A continuación analizaremos las siguientes directivas.

ServerType standalone o inetd

Por omisión esta directiva se encuentra con el valor de standalone aunque se puede utilizar el valor inetd, si se emplea el valor inetd entonces el servidor solo se ejecutara cuando haya alguna petición al puerto 80 de otra manera permanecerá inactivo, es recomendable que permanezca esta directiva con su valor por omisión que es standalone con este valor apache se encuentra activo y escuchando peticiones.

Standalone solo carga el servidor principal y los de apoyo quedan en espera y solo se activan según se demande, una característica de la opción standalone es que se tiempo de respuesta es bajo y con poco consumo de recursos.

Inetd se cargan todos los servidores de apoyo en memoria tiempo de respuesta alto (rápido), alto consumo de recursos.

ServerType standalone

ServerRoot establece la ruta del directorio en el que se almacenan donde se encuentran los archivos del servidor apache, el valor por omisión de esta directiva es /var/lib/apache.

En esta ruta "/var/lib/apache" es donde se encuentran los directorios donde se incluyen los archivos binarios y de configuración de apache, por ejemplo podemos encontrar en este lugar el directorio /conf, aquí es donde se encuentra el archivo httpd.conf, también se puede encontrar el directorio /bin donde se almacenan los binarios para ejecutar apache, al menos que durante la instalación se haya definido otra ruta el valor de la directiva ServerRoot es la siguiente.

ServerRoot "/var/lib/apache"

PidFile Contiene la ruta del directorio donde se almacena en número PID (identificador de proceso) para el servidor apache, este número se reserva para la ejecución de apache, ningún otro proceso podra utilizar este PID.

PidFile "/var/run/httpd.pid"

Timeout Esta directiva especifica el valor de expiración de la comunicación con el cliente, específicamente en el momento de que se produzca uno de estos eventos.

El tiempo total para recibir una solicitud **GET**.

El tiempo que hay entre los paquetes de una solicitud **PUT** o **POST**.
El tiempo que hay entre los **ACK** de transmisión de paquetes de respuesta.

Timeout 300

KeepAlive Establece la opción de servir mas de una solicitud en una misma conexión. Esta directiva tiene un valor de **On** lo que significa que esta activada, un valor de **Off** desactivara la opción.

Si una página contiene 3 imágenes no será necesario hacer 4 solicitudes al servidor, una para la página web y 3 para las imágenes, sino que con una sola solicitud basta para atender varias solicitudes con esto se acelera el tiempo de respuesta del servidor.

KeepAlive On

MaxKeepAliveRequest Determina el número máximo de solicitudes en una sola conexión. Esta directiva esta definida con un valor de 100 lo que significa que se aceptaran 100 para una sola conexión.

Un valor de 0 en esta directiva significa que no se tendrá un número limitado de conexiones.

MaxKeepAliveRequest 100

KeepAliveTimeout Especifica el número de segundos que van a transcurrir antes de cerrar una conexión con el cliente.

KeepAliveTimeout 15

MinSpareServers El número mínimo de procesos secundarios que pueden estar ejecutando.

MinSpareServers 5

MaxSpareServers El número máximo de procesos secundarios que pueden estar ejecutando.

MaxSpareServers 10

Si existen menos procesos de los que especifica la directiva **MinSpareServers** entonces se crearán más procesos, si están en ejecución mas procesos de los que establece la directiva **MaxSpareServers** entonces serán eliminados, con estas dos directivas se asegura que el servidor cuente con los procesos secundarios necesarios para aceptar peticiones de varios clientes.

StartServers Establece el número de servidores de apoyo o atención a peticiones directas que comienzan cuando inicia el servidor web. Este valor esta especificado en la directiva **MinSpareServers**.

StartServers 5

MaxClients Establece el número máximo de solicitudes simultaneas de cliente que van a ser atendidas. Esta directiva tiene un valor de 150 por omisión.

MaxClients 150

MaxRequestPerChild Número de solicitudes que va a servir un proceso secundario antes de que se cierre. Esta directiva tiene una valor de 0 que significa que nunca se cerrara y siempre estará atendiendo peticiones.

MaxRequestPerChild 0

Todas estas variables pueden ser modificadas acorde a las características del servidor.

4.3.3 Configuración básica del servidor.

Para realizar modificaciones al servidor apache se debe hacer referencia a la segunda sección Main server configuration del archivo de configuración httpd.conf, Como se ve en al figura siguiente.

```
### Section 2: 'Main' server configuration
#
# The directives in this section set up the values used by the 'main'
# server, which responds to any requests that aren't handled by a
# <VirtualHost> definition.  These values also provide defaults for
# any <VirtualHost> containers you may define later in the file.
#
# All of these directives may appear inside <VirtualHost> containers,
# in which case these default settings will be overridden for the
# virtual host being defined.
#
```

Figura 4.9 Sección 2 configuración del servidor

Esta sección al igual que la anterior tiene definidas directivas que se incluyen desde origen, aquí veremos solo algunas de ellas, lo primero que se hará antes de realizar configuraciones más complicadas será personalizar el servidor apache de tal manera que aparezcan algunas caracteristicas del servidor como el nombre del host y la dirección de correo electrónico del administrador.

La directiva Port le indica al servidor apache el puerto TCP/IP por el cual debe escuchar peticiones HTTP. El valor definido por omisión es el número 80 que es el puerto estándar para este tipo de conexiones.

Port 80

Las directivas **User** especifica el nombre del usuario y la directiva **Group** especifica el grupo con el que se ejecutará el servidor web, este usuario no debe tener ningún privilegio por cuestiones de seguridad del servidor, el grupo al que pertenece este

usuario debe tener las mismas características, por omisión tienen un valor de nobody, que es el usuario que se emplea para este propósito.

User nobody
Group nobody

La directiva **ServerAdmin** especifica el correo electrónico del administrador del servidor, esta directiva tiene un valor de **root@zap.Slackware.lan** se debe modificar por una cuenta valida de correo electrónico o por la cuenta de root pero se recomienda que sea una cuenta sin privilegios.

ServerAdmin admin@miservidor.com.mx

ServerName define el nombre del servidor, por lo regular aparece comentada, se debe quitar el comentario y sustituir lo que ahí aparece con el nombre del servidor.

ServerName miservidor.com.mx

DocumentRoot define la ruta del directorio que contiene el archivo de inicio (index.html) de la pagina principal, así como los archivos que se incluyen como ligas dentro del documento **index.html**.

DocumentRoot "/var/lib/apache/htdocs"

Este valor no debe ser modificado al menos que se quiera especificar una ruta distinta.

DirectoryIndex define el nombre del archivo escrito en html que se usa como pagina principal, es decir el archivo que se busca todo navegador web por omisión para mostrar un sitio de Internet, si es que no se especifica un nombre de archivo en la URL del servidor web.

Por ejemplo si le pedimos a un navegador web la pagina web del sitio www.yahoo.com entonces el navegador buscara el archivo index.html como primer opción para mostrar la página de inicio de este sitio, pero si especificamos la siguiente ruta www.yahoo.com/juegos.html entonces el navegador buscara el archivo juegos.html que hemos especificado.

DirectoryIndex Index.html

El valor para esta directiva es **index.html**, se recomienda que este valor no sea modificado al menos que se quiera especificar algo distinto.

Las modificaciones echas al archivo de configuración se deben ver como lo muestra la imagen siguiente, se guardan los cambios y solo falta reiniciar el servidor para que los cambios surtan efecto.

Section 2: 'Main' server configuration

```
#
# The directives in this section set up the values used by the 'main'
# server, which responds to any requests that aren't handled by a
# <VirtualHost> definition.  These values also provide defaults for
# any <VirtualHost> containers you may define later in the file.
#
# All of these directives may appear inside <VirtualHost> containers,
# in which case these default settings will be overridden for the
# virtual host being defined.
#

#
# If your ServerType directive (set earlier in the 'Global Environment'
# section) is set to "inetd", the next few directives don't have any
# effect since their settings are defined by the inetd configuration.
# Skip ahead to the ServerAdmin directive.
#

#
# Port: The port to which the standalone server listens. For
# ports < 1023, you will need httpd to be run as root initially.
#
Port 80

#
# If you wish httpd to run as a different user or group, you must run
# httpd as root initially and it will switch.
#
# User/Group: The name (or #number) of the user/group to run httpd as.
#  . On SCO (ODT 3) use "User nouser" and "Group nogroup".
#  . On HPUX you may not be able to use shared memory as nobody, and the
#    suggested workaround is to create a user www and use that user.
#  NOTE that some kernels refuse to setgid(Group) or semctl(IPC_SET)
#  when the value of (unsigned)Group is above 60000;
#  don't use Group nobody on these systems!
#
User nobody
Group nobody

#
# ServerAdmin: Your address, where problems with the server should be
# e-mailed.  This address appears on some server-generated pages, such
# as error documents.
#
ServerAdmin admin@miservidor.com.mx

#
```

```
# ServerName allows you to set a host name which is sent back to clients for
# your server if it's different than the one the program would get (i.e., use
# "www" instead of the host's real name).
#
# Note: You cannot just invent host names and hope they work. The name you
# define here must be a valid DNS name for your host. If you don't understand
# this, ask your network administrator.
# If your host doesn't have a registered DNS name, enter its IP address here.
# You will have to access it by its address (e.g., http://123.45.67.89/)
# anyway, and this will make redirections work in a sensible way.
#
# 127.0.0.1 is the TCP/IP local loop-back address, often named localhost. Your
# machine always knows itself by this address. If you use Apache strictly for
# local testing and development, you may use 127.0.0.1 as the server name.
#
ServerName miservidor.com.mx

#
# DocumentRoot: The directory out of which you will serve your
# documents. By default, all requests are taken from this directory, but
# symbolic links and aliases may be used to point to other locations.
#
DocumentRoot "/var/lib/apache/htdocs"

#
# Each directory to which Apache has access, can be configured with respect
# to which services and features are allowed and/or disabled in that
# directory (and its subdirectories).
#
# First, we configure the "default" to be a very restrictive set of
# permissions.
#
<Directory />
    Options FollowSymLinks
    AllowOverride None
</Directory>

#
# Note that from this point forward you must specifically allow
# particular features to be enabled - so if something's not working as
# you might expect, make sure that you have specifically enabled it
# below.
#
#
# This should be changed to whatever you set DocumentRoot to.
#
<Directory "/var/lib/apache/htdocs">
```

```
#
# This may also be "None", "All", or any combination of "Indexes",
# "Includes", "FollowSymLinks", "ExecCGI", or "MultiViews".
#
# Note that "MultiViews" must be named *explicitly* --- "Options All"
# doesn't give it to you.
#
    Options Indexes FollowSymLinks MultiViews

#
# This controls which options the .htaccess files in directories can
# override. Can also be "All", or any combination of "Options", "FileInfo",
# "AuthConfig", and "Limit"
#
    AllowOverride None

#
# Controls who can get stuff from this server.
#
    Order allow,deny
    Allow from all
</Directory>

#
# UserDir: The name of the directory which is appended onto a user's home
# directory if a ~user request is received.
#
<IfModule mod_userdir.c>
    UserDir public_html
</IfModule>

#
# Control access to UserDir directories.  The following is an example
# for a site where these directories are restricted to read-only.
#
#<Directory /home/*/public_html>
#    AllowOverride FileInfo AuthConfig Limit
#    Options MultiViews Indexes SymLinksIfOwnerMatch IncludesNoExec
#    <Limit GET POST OPTIONS PROPFIND>
#        Order allow,deny
#        Allow from all
#    </Limit>
#    <LimitExcept GET POST OPTIONS PROPFIND>
#        Order deny,allow
#        Deny from all
#    </LimitExcept>
```

```
#</Directory>

#
# DirectoryIndex: Name of the file or files to use as a pre-written HTML
# directory index.  Separate multiple entries with spaces.
#
<IfModule mod_dir.c>
    DirectoryIndex index.html
</IfModule>
```

Figura 4.10 Sección 2 directivas de configuración.

Cada directiva puede ser personalizada acorde a las características del servidor, a continuación se muestra como reiniciar el servidor apache para que relea el archivo de configuración httpd.conf.

4.3.4 Iniciar y detener el servidor apache.

Cuando se realizan modificaciones en el archivo de configuración es necesario detener y arrancar el servidor para que los cambios surtan efecto, no es necesario reiniciar el sistema por completo solo es necesario reiniciar el servidor apache con el comando apachectl que se encuentra en el directorio /var/lib/apache/sbin.

El comando apachectl cuenta con diferentes opciones:

#apachectl [opción]

Donde opción puede ser:

stop
start
restart
status

la opción **stop** detiene el servidor web y la opción **start** lo arranca, la opción **restart** hace estas dos funciones a la vez detiene y arranca el servidor al mismo tiempo, la opción status nos sirve para saber el estado del servidor si esta detenido o corriendo.

Para hacer valida la configuración anterior es necesario detener y arrancar el servidor de la siguiente forma:

#/var/lib/apache/sbin/apachectl restart

Para comprobar la configuración se ejecuta cualquier navegador web y se teclea la IP o nombre del servidor junto con algún nombre de archivo que no exista para verificar el valor de las directivas ServerName y ServerAdmin.

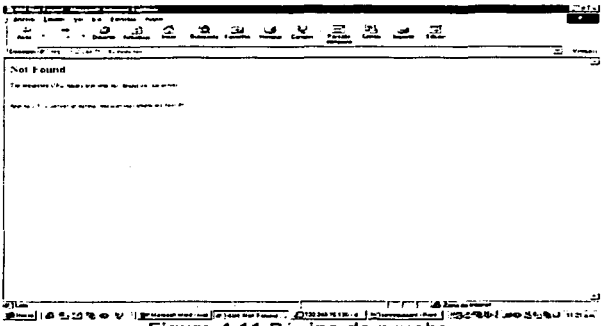


Figura 4.11 Página de prueba.

4.4 Página principal.

El servidor web apache almacena los archivos con extensión html dentro del directorio /var/lib/apache/htdocs, la pagina principal de un sitio web en Internet se almacena en este lugar y demás archivos (ligas) que haga referencia la pagina principal.

El nombre del archivo para la página principal debe ser único y ningún otro archivo debe contener este nombre, el servidor web siempre buscará este archivo cada vez que un cliente pida la pagina principal de un sitio web, el archivo index.html es el archivo que se define en las directivas del servidor web como el archivo principal de un sitio web.

```
<IfModule mod_dir.c>
   DirectoryIndex index.html
</IfModule>
```

Figura 4.12 Archivo index.html

El archivo index.html debe contener permisos de lectura, escritura y ejecución para el dueño y tan solo permisos de ejecución tanto para el grupo como para los demás para que cualquier cliente pueda observar su contenido desde cualquier navegador pero no debe contener permisos de escritura para evitar modificaciones a este archivo.

Una vez diseñada la página principal del centro y después de haber realizado algunas pruebas la página principal quedo lista y a disposición del publico en general. A continuación se muestra una imagen con la página principal (hermes.mascarones.unam.mx) del Centro Mascarones.

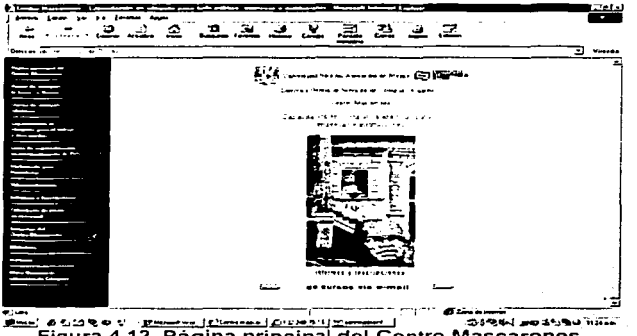


Figura 4.12. Página principal del Centro Mascarones.

CAPÍTULO 5
SEGURIDAD

5. Seguridad en el servidor.

La seguridad es una parte importante en un sistema de computo aunque se dice que no existe ningún sistema que sea 100% seguro, existen una gran gama de libros y artículos en Internet sobre el tema de seguridad que son gulas acerca de cómo implantar seguridad en sistemas de computo, para que este sea menos susceptible a ataques externos o internos de personas mal intencionadas.

Este capitulo trata sobre como implantar seguridad en un sistema Links, pero va mas enfocado a como implantar seguridad en la versión 7.1 de Links Slackware y en su servidor web apache, se mencionara solo el trabajo que se realizo para mejorar la seguridad del servidor web Hermes, algunas de estas mejoras fueron implantadas a los demás servidores Linux.

5.1. Conexiones remotas.

Para fines de los cursos que se imparten en el Centro Mascarones se utilizan los servicios de telnet y ftp como parte de los temarios de los cursos de UNIX.

De acuerdo con algunos libros de seguridad no es recomendable utilizar los servicios de telnet y ftp para la administración remota de servidores ya que estos protocolos son bastante inseguros tan es así que telnet no permite realizar una conexión remota con un servidor utilizando la cuenta root.

Debido a la inexperiencia de algunos administradores utilizan telnet y ftp para administrar su servidor ya que estos protocolos son estándar en casi todos los sistemas, son fáciles de utilizar y en cualquier lugar se puede encontrar documentación acerca de su uso, y en muchas ocasiones por desconocimiento del propio administrador no sabe que tiene habilitado este servicio.

Para solucionar este problema se tomo la decisión de utilizar el programa SSH (Secure Shell), este programa utiliza diferentes algoritmos de encriptación para hacer segura una conexión remota, la versión 7.1 de Slackware no cuenta con una versión instalada de este programa, así es que se tuvo que decidir entre dos diferentes programas que realizan el mismo trabajo para ser instalado en Hermes, Openssh una versión libre del programa ssh y desarrollada por las personas de Openbsd un unix hecho exclusivamente para seguridad y el programa original ssh desarrollado por la empresa con el mismo nombre.

La historia de ambos programas viene ligada una de otra, el protocolo ssh fue creado por un finlandés, en sus orígenes ssh era de código abierto y su distribución era libre sin ningún tipo de restricciones, al paso del tiempo ssh fue ganando adeptos y mejores características hasta que su creador decide vender el software bajo una licencia comercial, esto causo que se creara un grupo aparte para desarrollar un protocolo similar a ssh pero que fuera libre de todo a todo este proyecto recibió el nombre de Openssh y se distribuye bajo una licencia no comercial que es bastante flexible.

Actualmente esta disponible la versión 3.1 del programa ssh y la versión 3.2.3p1 de Openssh. La versión 3.1 de ssh incluye una versión comercial y otra no comercial que incluye a instituciones educativas entre ellas las universidades, mientras que openssh es libre para cualquier tipo de uso incluyendo el comercial.

Algunas caracteristicas de estos programas son:

Las características de SSH Secure Shell incluyen:

- Protección a todas las contraseñas y datos.
- Reemplaza a protocolos como telnet, rlogin, rsh, rcp y ftp.
- Integra transferencia segura de archivos y copia de archivos
- Incluye una interfaz gráfica de usuario para entornos Windows.
- Autenticación automática de usuarios, no envía claves en claro.
- Autenticación por ambos lados de la conexión tanto del cliente como del servidor, son autenticados para prevenir el identity spoofing, caballos de Troya, etc.
- Protección a sesiones X11.
- Encriptación y compresión de datos para seguridad y velocidad.
- Múltiple construcción de métodos de autenticación, incluyendo passwords, llaves publicas, securid y host basado en autenticación.
- Soporte para PKI y hardware token.
- Múltiples algoritmos de encriptamiento incluyendo 3DES, Blowfish, Twofish y AES.

Las caracteristicas de Openssh son:

- Proyecto de Código Abierto
- Licencia Libre
- Cifrado Fuerte (3DES, Blowfish)
- Reenvio por X11 (cifra el tráfico de X Window System)
- Reenvio por Puertos (canales cifrados por protocolos de legado)
- Autenticación Fuerte (Clave Pública, Contraseña de un sólo uso y Autenticación con Kerberos)
- Reenvio por Agente (ingreso único)
- Interoperabilidad (Conforme con las Regulaciones del Protocolo SSH 1.3, 1.5, y 2.0)
- Soporte para cliente y servidor de SFTP en los protocolos SSH1 y SSH2.
- Pases de Ticket de Kerberos y AFS
- Compresión de Datos

Otra característica importante de estos dos programas es su interoperabilidad con diferentes plataformas a continuación se muestran algunos de los sistemas operativos en donde ssh y Openssh pueden trabajar.

Plataformas que soporta SSH:

- Microsoft Windows 95/98
- Microsoft Windows NT 4.0/2000
- Sun Solaris
- HP/UX
- AIX
- Linux (Slackware, Redhat, Suse y Caldera)

Plataformas que soporta Openssh.

- OpenBSD
- Debian Linux
- FreeBSD
- Suse Linux
- Redhat Linux
- Mandrake Linux
- BSDi BSD/OS
- NetBSD
- Computone
- Conectiva Linux
- Slackware Linux
- Caldera OpenLinux
- Stallion
- Rock Linux
- Cygwin
- servidor y pasarela e-smith
- Engarde Linux
- MacOS X Version 10.1
- HP Procurve Switch 4108GL and 2524/2512
- IBM AIX
- Gentoo Linux
- Gwynux/Toadware Linux

Los dos programas se pueden descargar directamente del sitio que pertenece a cada uno de ellos para ssh se puede descargar de www.ssh.com y openssh se puede descargar de www.openssh.org , los paquetes que son descargados contiene una extensión tar.gz que es la extensión por omisión de los paquetes para plataformas tipo Unix.

A continuación se incluyen algunas características de uso de este protocolo en Internet, estas se obtuvieron de la pagina web que es un proyecto similar al de netcraft, publica con carácter mensual estadísticas de uso e implementación del protocolo ssh en Internet, esto se puede apreciar mas gráficamente en las siguientes imágenes.

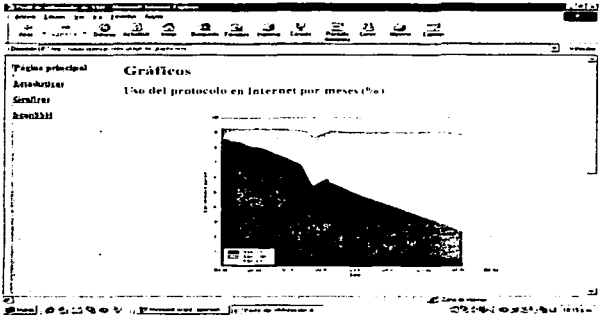


Figura 5.1 Uso del protocolo ssh en Internet por meses.

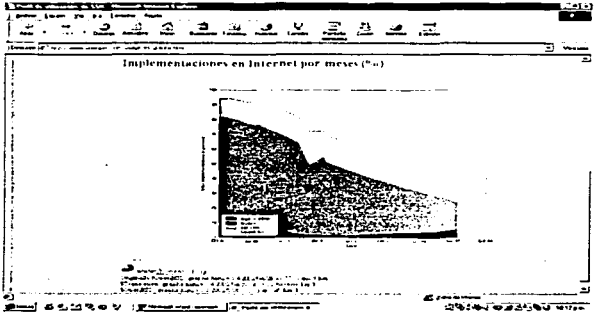


Figura 5.2 implementación de ssh por meses.

También se obtuvieron de este mismo sitio algunas tablas comparativas sobre los diferentes programas que utilizan el protocolo ssh y cual es el que más se utiliza.

Internet en abril de 2002. Muestra: 2.4 millones de direcciones aleatorias.

Versiones del protocolo:

0.2%	1.3
22.7%	1.5
65.8%	1.99
11.3%	2.0

Tabla 5.1 Versiones de SSH mas utilizadas.

Dæmons de SSH:

0.4%	Cisco-1.5
3.1%	OpenSSH-1.5
59.0%	OpenSSH-1.99
4.7%	OpenSSH-2.0
1.3%	Other-1.5
1.3%	Other-1.99
2.1%	Other-2.0
0.2%	SSH-1.3
17.9%	SSH-1.5
5.5%	SSH-1.99
4.5%	SSH-2.0

Tabla 5.2 Versiones del Demonio SSH en Internet.

La forma de instalar cada paquete es muy similar y en realidad resulta sencilla la instalación, toda la documentación para la instalación de los paquetes viene incluida en los archivos que se descargan o si es necesario se puede consultar directamente la pagina web de los fabricantes.

Para instalar SSH es necesario obtener una versión del software la cual se puede obtener de las ligas que se mencionaron con anterioridad. Después de haber obtenido el paquete de SSH versión**2.4.0** que en este momento es la versión actualizada del protocolo, procedemos a desempaquetarlo, con las siguientes instrucciones,

Antes de instalar el paquete se recomienda que cuando se instale un paquete en Linux, este sea desempaquetado en el directorio /usr/local que es el directorio donde se instalan los paquetes externos que se consiguen en algún otro sitio (Internet, CDS, etc.) y que no están incluidos en los CDS de instalación de Linux.

Primero se descomprime el paquete.

Sgunzip ssh-2.4.0.tar.gz

Después utilizamos el comando tar para extraer el contenido del archivo.
Star -xvf ssh-2.4.0.tar

En este punto obtendremos un directorio **./ssh-2.4.0/** sobre el punto donde desempaquetamos (/usr/local). A continuación se citan los pasos necesarios para obtener los binarios de **ssh**, compilando los fuentes:

A diferencia de otras herramientas, **ssh** no requiere de editar los archivos **Makefile**, la configuración la realizamos a través de los parámetros que le pasemos al script configure, este script verificara entre otras cosas el tipo de SO que estamos utilizando, de esta manera el script creara un makefile personalizado para nuestro sistema..

Dentro del directorio**./ssh-2.4.0/** encontraremos el script **configure**, el cual tiene los siguientes argumentos válidos:

--prefix=PREFIX Donde se instalan los binarios por default **/usr/local.**

--exec_prefix=PREFIX Donde se instalarán los ejecutables, por default el mismo que prefix.

--with-clientsecurId Habilita el soporte para la autenticación SecurID del cliente.

--without-idea No incluir **IDEA.**

--with-tis=PATH Soporte a mecanismo de autentificación Tis authsrv.
--with-etcdir=PATH Donde se instalaran los archivos de configuración por default /etc.

--with-libwrap=[PATH] Usa **libwrap** (tcp_wrappers) y **identd.**

--enable-verbose-warnigs Habilita la bandera Wall al compilador gcc.
Si nuestra intención es lograr que **tcp-wrapper** lleve un control sobre los accesos realizados con ssh, se necesitará la bandera **--with-libwrap**.

$./configure --with-libwrap Como segundo paso necesitamos compilar los binarios de **SSH**. Para esto basta ejecutar:

$make

Por ultimo ejecutamos el comando make install para instalar **SSH** y obtener los archivos ejecutables y los archivos de configuración del sistema:

$make install

Ya que obtuvimos los binarios del sistema procedemos a configurar el sistema para poder ejecutar el demonio del servidor de ssh y permitir accesos por el puerto por default de **SSH** (puerto 22).
Editamos el archivo **/etc/inetd.conf** para incluir la siguiente linea si se tiene demonio de **tcp-wrapper.**

ssh tcp root nowait /usr/local/sbin/tcpd /usr/local/sbin/sshd2 -i

Si no se tiene habilitado **tcp-wrapper.**

ssh root nowait /usr/local/sbin/sshd /usr/local/sbin/sshd2 –i

Enseguida Editamos el archivo **/etc/services** y habilitar el puerto de **Secure Shell**, con las siguientes lineas:

ssh 22/tcp
ssh 22/udp

Por último reiniciamos el demonio de **inetd**. Obteniendo el process_id de inetd con la siguiente instrucción.

$ps -fea | grep inetd

Enviamos la señal **HUP** junto con el process_id por medio del comando kill.
$kill -HUP process_id

Después de todo esto el sistema debe responder las peticiones de conexión por **Secure Shell**.Para estar seguros que **SSH** funciona realizamos una prueba de conexión:

$ssh –l leopoldo IP

Si todo sale bien el servidor responderá pidiéndonos un password para permitirnos el acceso repodemos escribiendo nuestra contraseña y listo **SSH** esta funcionando.

Cada programa tiene su propio historial de seguridad es recomendable que cuando se instala un nuevo programa se obtenga la ultima versión, de esta manera se puede asegurar que los errores detectados en otras versiones están corregidos, esto no implica que la nueva versión no contenga errores.

Dos lugares que son recomendables consultar para mantenerse informado acerca de las vulnerabilidades que están surgiendo continuamente en las diferentes aplicaciones son www.securityfocus.com y la pagina del CERT www.cert.org para estar actualizado acerca de virus, bugs y ataques.

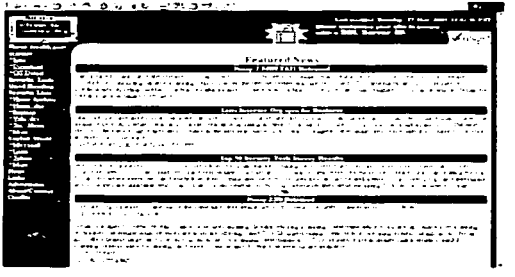
Después de leer algunos avisos de seguridad utilizando estos programas se decidió por instalar ssh 3.0.1 en su versión no comercial ya que openssh es gratuito y viene incluido en la mayoria de las distribuciones Linux pero se encuentra todavia en desarrollo lo que hace que se encuentren continuamente bugs de seguridad y hay que estar constantemente poniendo parches para corregir estos errores.

5.2. Escaneo de puertos.

Una de las formas en como los hackers buscan lugares por donde se pueden introducir es a través de los puertos que están ofreciendo algún tipo de servicio, esto se hace a través de programas para escaneo de puertos, estos programas lo que hacen es enviar paquetes a través de la red al host que se quiere atacar a diferentes números de puertos, si el puerto esta abierto entonces contestara cada vez que llegue alguna petición, en caso de que el paquete enviado por el programa de escaneo de puertos es contestado entonces quiere decir que ese puertos se encuentra abierto y escuchando.

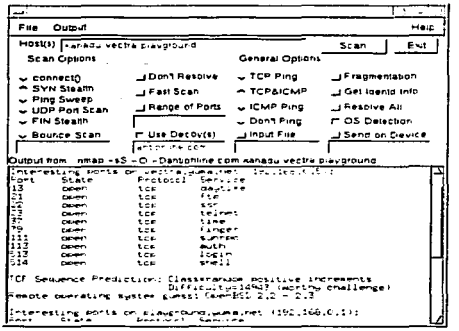
Es muy común este tipo de ataques hacia host remotos, para evitar ser atacados por un puerto que no sé este ocupando es recomendable cerrarlo y dejar solo abiertos a aquellos que son necesarios, pero antes de cerrar puertos es necesario saber cuales son los puertos que se tienen abiertos, para este propósito se utilizara un programa de escaneo de puertos, en realidad este tipo de programas se crearon para ayudar al administrador de sistemas a detectar aquellos puertos abiertos en su sistema y en realidad son herramientas que auxilian en la administración de un servidor pero los hackers sacan provecho de estas herramientas para sus propios intereses.

La herramienta que se utilizo para el escaneo de puertos o barrido de puertos como también se le conoce, es la herramienta nmap, esta herramienta es la mejor en su tipo ya que reúne todos los tipos de escaneo de puertos que existen, además de otra caracteristica especial que es la de detectar el sistema operativo del sistema que se esta escaneando, existen versiones para los diferentes unix incluyendo Linux esta herramienta se puede descargar de www.insecure.org/nmap dentro de la pagina principal viene incluida la forma en como se instala y configura esta herramienta.



5.3 Página principal de www.insecure.org

Hay que tomar en cuenta que esta herramienta es para uso de los administradores de sistemas y para eso fue creada, se puede trabajar de dos formas con nmap a través de la linea de comandos o utilizando una interfaz gráfica que es mucho más fácil de utilizar pero no tan interesante como la versión en modo texto.



5.4 interfaz gráfica de nmap.

5.5 nmap desde la línea de comandos.

Con la ayuda de nmap se puede detectar los puertos abiertos en el sistema, ahora solo falta cerrar esos puertos, la mayoria de los servicios se encuentran declarados dentro un archivo de configuración dentro del directorio /etc este archivo contiene las

caracteristicas de los diferentes servicios que se activan cuando arranca el sistema, el archivo services contiene toda esta información el contenido de este archivo es el siguiente:

```
# /etc/services:
# $Id: services,v 1.17 2001/02/28 20:11:31 notting Exp $
#
# Network services, Internet style
#
# Note that it is presently the policy of IANA to assign a single well-known
# port number for both TCP and UDP; hence, most entries here have two entries
# even if the protocol doesn't support UDP operations.
# Updated from RFC 1700, ``Assigned Numbers'' (October 1994).  Not all ports
# are included, only the more common ones.
#
# The latest IANA port assignments can be gotten from
#       http://www.isi.edu/in-notes/iana/assignments/port-numbers
# The Well Known Ports are those from 0 through 1023.
# The Registered Ports are those from 1024 through 49151
# The Dynamic and/or Private Ports are those from 49152 through 65535
#
# Each line describes one service, and is of the form:
#
# service-name  port/protocol  [aliases ...]   [# comment]

tcpmux          1/tcp                           # TCP port service multiplexer
tcpmux          1/udp                           # TCP port service multiplexer
rje             5/tcp                           # Remote Job Entry
rje             5/udp                           # Remote Job Entry
echo            7/tcp
echo            7/udp
discard         9/tcp           sink null
discard         9/udp           sink null
systat 11/tcp users
systat 11/udp       users
daytime         13/tcp
daytime         13/udp
qotd            17/tcp quote
qotd            17/udp          quote
msp             18/tcp                  # message send protocol
msp             18/udp                          # message send protocol
chargen         19/tcp ttytst source
chargen         19/udp          ttytst source
ftp-data        20/tcp
ftp-data        20/udp
ftp             21/tcp
ftp             21/udp
```

```
ssh             22/tcp                  # SSH Remote Login Protocol
ssh             22/udp                  # SSH Remote Login Protocol
telnet          23/tcp
telnet          23/udp
# 24 - private mail system
smtp            25/tcp          mail
smtp            25/udp          mail
time            37/tcp timserver
time            37/udp          timserver
rlp             39/tcp resource     # resource location
rlp             39/udp          resource    # resource location
nameserver      42/tcp name             # IEN 116
nameserver      42/udp          name        # IEN 116
nicname         43/tcp          whois
nicname         43/udp          whois
tacacs49/tcp                    # Login Host Protocol (TACACS)
tacacs49/udp                    # Login Host Protocol (TACACS)
re-mail-ck      50/tcp                  # Remote Mail Checking Protocol
```

Figura 5.6 Extracto del archivo services.

La forma de cancelar un servicio en /etc/services es muy simple solo basta con poner un comentario detrás de la línea donde se define un servicios y listo, el símbolo que se utiliza para poner un comentario es el símbolo # cuando el sistema arranque y lea la configuración del archivo /etc/services entenderá que las líneas con este símbolo son comentarios y no activara ese servicio, para el servidor Hermes solo se tienen activados los servicios de ssh, WWW, pop3 y smtp

Una forma de evitar el escaneo de puertos es utilizar una herramienta anti-escaneo, un programa muy buena para este propósito es portsentry desarrollado por psionic, cuando portsentry detecta un escaneo de puertos por medio de tcp-wrapper reacciona bloqueando la dirección ip que realizo el scaneo agregando una línea dentro del archivo /etc/hosts.deny y otra línea dentro de la bitácora del sistema, podemos utilizar un software de filtrado de paquetes como iptables, para asegurarnos que el atacante no pueda conectarse a ninguno de los puertos de nuestro equipo ni que estos le respondan cuando este hace alguna petición de algún tipo como por ejemplo un ping, porsentry tiene varias modalidades avanzadas de trabajo toda la información de cómo trabaja porsentry se puede obtener de www.psionic.com/products/ o junto con el archivo que se descarga.

Se puede complementar la seguridad del sistema Linux con la ayuda de comandos y programas que vienen incluidos dentro del SO comandos como last, w, netstat, traceroute y herramientas como tripwire, tcpwrappers pueden auxiliar al administrador de sistemas en la tarea de la administración del servidor.

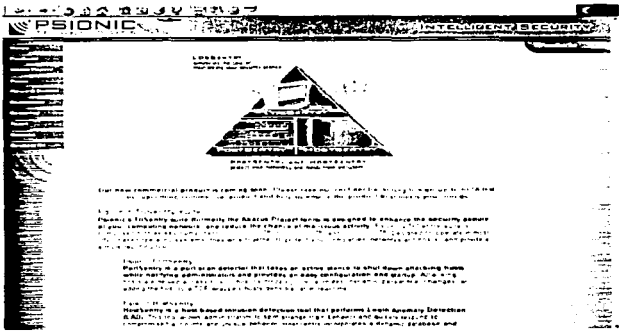


Figura 5.7 Pagina principal de Psionic.

5.3 Seguridad en apache.

Cada vez que un navegador web hace una conexión con el servidor crea un registro en las bitácoras del sistema, estas bitácoras contiene toda la información acerca de la conexión que se realizo con el servidor informando al administrador del sitio como ocurrió esta conexión.

Las bitácoras son el medio por el cual el administrador de un sitio web puede obtener información acerca de las conexiones que se han realizado al sitio web, estas bitácoras tienen un formato especial el cual es muy fácil de entender por ejemplo en la parte de abajo se tiene un extracto de las bitácoras del servidor Hermes.

```
66.196.73.21 - - [01/Jun/2002:00:47:35 -0500] "GET /mascarones/informes.html HTTP/1.0" 304
66.28.250.172 - - [01/Jun/2002:00:58:23 -0500] "GET /robots.txt HTTP/1.0" 404 286
66.28.250.172 - - [01/Jun/2002:00:58:24 -0500] "GET /temarios/Photoshop.html HTTP/1.0" 200
200.66.121.149 - - [01/Jun/2002:01:04:06 -0500] "GET /mascarones/act.html HTTP/1.1" 200
200.66.121.149 - - [01/Jun/2002:01:04:09 -0500] "GET /images/win.gif HTTP/1.1" 200 1041
```

Figura 5.8 Extracto del archivo error_log

Es importante para un administrador que maneje apache en su sitio web, saber leer cada una de estas lineas, cada linea esta dividida en 6 campos que son la IP del

cliente, Fecha, Hora, el método que se utilizo (GET, POST o HEAD) y la ruta del archivo que se solicito, código de estado del servidor, numero de bytes transferidos.

El servidor apache maneja una tabla de códigos de estado que informan acerca del estado del servidor web, estos códigos están formado por tres dígitos y cada código define un estado del servidor, cuando un cliente realiza una petición al servidor web y es satisfactoria la respuesta del servidor, entonces se genera un código 200 que significa que la conexión finalizo con éxito. A continuación se muestra una breve descripción de la tabla de códigos utilizada por apache.

TIPO	CODIGO
Informativo	100-109
Petición Correcta	200-299
Redirección de petición	300-399
Petición incorrecta	400-499
Error en el servidor	500-599

Tabla 5.3 Códigos de estado del servidor apache.

El servidor apache maneja dos tipos de bitácoras la primera bitácora guarda las conexiones exitosas que ha tenido el servidor web. Mientras que la segunda almacena los errores o conexiones fallidas.

Estas dos bitácoras manejan el mismo formato el archivo que almacena las conexiones exitosas recibe el nombre de access_log y el archivo que guarda las conexiones fallidas es el archivo error_log, la ubicación de las bitácoras de apache en Slackware 7.1 se encuentran en el directorio /var/log, en otras versiones de Linux la ubicación es similar.

Un ejemplo del contenido del archivo error_log se muestra a continuación.

```
[Sat Jun  1 00:00:04 2002] [notice] Apache/1.3.12 (Unix) PHP/4.1.1 configured -- resuming normal operations
[Sat Jun  1 00:58:23 2002] [error] [client 66.28.250.172] File does not exist: /var/lib/apache/htdocs/robots.txt
[Sat Jun  1 01:20:30 2002] [error] [client 63.84.238.133] File does not exist: /var/lib/apache/htdocs/temarios/CristalRepor$
[Sat Jun  1 01:20:35 2002] [error] [client 63.84.238.133] File does not exist: /var/lib/apache/htdocs/temarios/CristalRepor$
```

Figura 5.9 Contenido archivo error_log

Cada vez que se genera un error en apache una nueva línea es añadida en el archivo error_log, la información que se muestra en este archivo es muy similar a la mostrada en el archivo access_log, este archivo divide la información en varios campos donde se incluyen dia, mes, año y hora en que ocurrió el error, la dirección IP del cliente, un mensaje de error y la ruta del archivo que no se encontró.

Este tipo de bitácoras tienden a crecer bastante ya que cada vez que hay una conexión al servidor son generadas varias líneas en estos archivos, debido ha esto se recomienda de vez en cuando depurar bitácoras, una forma seria respaldar las bitácoras en una ruta diferente, eliminar los archivos access_log, error_log y sustituirlos por archivos vacios con el mismo nombre y permisos.

Esta tarea es recomendable realizarla una vez por semana o por lo menos cada fin de mes, pero seria bastante engorroso y aburrido realizar esta tarea en una fecha especifica sin contar el olvido del administrador, se recomienda también para esta tarea escribir un pequeño programa en shell que realice las tareas de respaldar, borrar, crear los archivos de bitácora, además de realizar estas atareas cada cierto tiempo, esta ultima tarea se puede realizar programando un cron.

Además se puede utilizar un programa de análisis de bitácoras como webalizer, existen muchos mas programas para el análisis de bitácoras pero este es el mejor en la categoría de software libre además de que se encuentra instalada una versión de este software en diferentes distribuciones Linux, webalizer permite que los reportes se generen en diferentes idiomas incluyendo el español.

Para instalar webalizer necesitamos obtener una copia del software que podemos descarga desde www.webalizer.com y tener cuenta de root, La versión que esta disponible en este momento es **webalizer-2.01-10.src.tgz** una vez que hemos obtenido una copia es hora de comenzar con la instalación. Lo primero que debemos hacer es colocar el archivo dentro del directorio **/usr/local**:

Smv webalizer-2.01-10.src.tgz /usr/local

Una vez echo lo anterior procedemos a descomprimir y extraer el contenido del archivo:

Star zxvf webalizer-2.01-10.src.tgz

En este punto obtendremos un directorio.**/webalizer-2.01-10/** sobre el punto donde desempaquetamos (/usr/local). Antes de comenzar con la instalación de webalizer es necesario instalar la librería gráfica gd la cual es necesaria para generar las imágenes que se muestran cuando se generan los reportes con webalizer, esta librería la podemos obtener de http://www.boutell.com/gd, la versión actualizada de gd es **gd-2.0.11.tar.gz**, descargamos una copia del sitio antes mencionado y ahora es momento de instalar gd a continuación se muestran los pasos para instalar gd:

Primero se descomprime y extraen los archivos del paquete gd:

Star zxvf gd-2.0.11.tar.gz

Entramos al directorio que se genero con la instrucción anterior y ahí ejecutamos la siguiente instrucción, para personalizar el paquete a nuestro sistema:

$./configure

Ejecutamos la instrucción make para compilar el paquete:
$make

por ultimo tecleamos make install para instalar los binarios, archivos de configuración y documentación de gd:

$make install

Una vez instalada la libreria grafica gd ahora se procede a mencionar los pasos para compilar e instalar webalizer:

Salimos del directorio gd si es que aun continuamos ahí e ingresamos al directorio **/webalizer-2.01-10/.** La razón por la cual descargamos una versión con los fuentes de webalizer es para personalizar este paquete de acuerdo con nuestras necesidades.

El paquete webalizer soporta diferentes idiomas entre ellos el español además podemos activar una opción especial de webalizer que es la de poder utilizar un DNS para resolver dominios de Internet de esta manera podemos tener mas información acerca de quien nos visita.

Primero personalizamos webalizer con la siguiente instrucción:

$./configure –with-language=spanish –enable-dns

Ahora compilamos el paquete de la misma forma que gd ejecutamos la instrucción make:

$make

Instalamos los binarios, archivos de configuración y documentación:

$make install

Antes de crear los reportes de nuestro sitio web con la ayuda de webalizer es necesario editar el archivo webalizer.conf que se instalo en el directorio /etc, entre otras cosas tenemos que indicarle a webalizer donde se encuentran los logs del sistema, para que pueda generar los reportes.

A continuación se mencionan las directivas que son necesario modificar y los valores que pueden adoptar

LogFile define el archivo log del servidor web a utilizar, sino se especifica aquí se puede especificar desde la linea de comandos.

LogFile /var/log/access.log

OutputDir define el lugar donde se quiere almacenar los reportes que genera webalizer, por omisión debe ser un directorio y este debe estar ubicado en el mismo lugar que las paginas web.

OutputDir /var/lib/apache/htdocs/reportes

HistoryName permite especificar el nombre del archivo historial de webalizer, el archivo historial contiene los datos que se levantan durante los 12 meses del año, con la información almacenada en este archivo se genera el archivo index.html, por omisión este archivo se llama webalizer.hist y se ubica en el mismo directorio que los reportes.

HistoryName /var/lib/apache/htdocs/reportes/webalizer.hist

ReportTitle es el texto que se despliega como titulo en la pagina principal, aparece junto a una cadena que por lo regular esta en ingles "Usage Statistics for" al final de esta cadena aparece el nombre de nuestro servidor.

ReportTitle Usage Statistics for miservidor

HostName define el nombre del host para el reporte.

HostName miservidor.com.mx

Una vez realizados estos cambios el archivo de configuración webalizer.conf se debe ver de la siguiente manera.

```
#
# Sample Webalizer configuration file
# Copyright 1997-2000 by Bradford L. Barrett (brad@mrunix.net)
#
# Distributed under the GNU General Public License.  See the
# files "Copyright" and "COPYING" provided with the webalizer
# distribution for additional information.
#
# This is a sample configuration file for the Webalizer (ver 2.01)
# Lines starting with pound signs '#' are comment lines and are
# ignored.  Blank lines are skipped as well.  Other lines are considered
# as configuration lines, and have the form "ConfigOption  Value" where
# ConfigOption is a valid configuration keyword, and Value is the value
# to assign that configuration option.  Invalid keyword/values are
# ignored, with appropriate warnings being displayed.  There must be
# at least one space or tab between the keyword and its value.
#
# As of version 0.98, The Webalizer will look for a 'default' configuration
# file named "webalizer.conf" in the current directory, and if not found
# there, will look for "/etc/webalizer.conf".
```

```
# LogFile defines the web server log file to use.  If not specified
# here or on on the command line, input will default to STDIN.  If
# the log filename ends in '.gz' (ie: a gzip compressed file), it will
# be decompressed on the fly as it is being read.

LogFile      /var/log/access_log

# LogType defines the log type being processed.  Normally, the Webalizer
# expects a CLF or Combined web server log as input.  Using this option,
# you can process ftp logs as well (xferlog as produced by wu-ftp and
# others), or Squid native logs.  Values can be 'clf', 'ftp' or 'squid',
# with 'clf' the default.
#LogType     clf

# OutputDir is where you want to put the output files.  This should
# should be a full path name, however relative ones might work as well.
# If no output directory is specified, the current directory will be used.

OutputDir    /var/lib/apache/htdocs/reportes

# HistoryName allows you to specify the name of the history file produced
# by the Webalizer.  The history file keeps the data for up to 12 months
# worth of logs, used for generating the main HTML page (index.html).
# The default is a file named "webalizer.hist", stored in the specified
# output directory.  If you specify just the filename (without a path),
# it will be kept in the specified output directory.  Otherwise, the path
# is relative to the output directory, unless absolute (leading /).

HistoryName        /var/lib/apache/htdocs/reportes/webalizer.hist

# Incremental processing allows multiple partial log files to be used
# instead of one huge one.  Useful for large sites that have to rotate
# their log files more than once a month.  The Webalizer will save its
# internal state before exiting, and restore it the next time run, in
# order to continue processing where it left off.  This mode also causes
# The Webalizer to scan for and ignore duplicate records (records already
# processed by a previous run).  See the README file for additional
# information.  The value may be 'yes' or 'no', with a default of 'no'.
# The file 'webalizer.current' is used to store the current state data,
# and is located in the output directory of the program (unless changed
# with the IncrementalName option below).  Please read at least the section
# on Incremental processing in the README file before you enable this option.

#Incremental no

# IncrementalName allows you to specify the filename for saving the
```

```
# incremental data in.  It is similar to the HistoryName option where the
# name is relative to the specified output directory, unless an absolute
# filename is specified.  The default is a file named "webalizer.current"
# kept in the normal output directory.  If you don't specify "Incremental"
# as 'yes' then this option has no meaning.

#IncrementalName webalizer.current

# ReportTitle is the text to display as the title.  The hostname
# (unless blank) is appended to the end of this string (seperated with
# a space) to generate the final full title string.
# Default is (for english) "Usage Statistics for".

ReportTitle   Usage Statistics for miservidor

# HostName defines the hostname for the report.  This is used in
# the title, and is prepended to the URL table items.  This allows
# clicking on URL's in the report to go to the proper location in
# the event you are running the report on a 'virtual' web server,
# or for a server different than the one the report resides on.
# If not specified here, or on the command line, webalizer will
# try to get the hostname via a uname system call.  If that fails,
# it will default to "localhost".

HostName      miservidor.com.mx

# HTMLExtension allows you to specify the filename extension to use
# for generated HTML pages.  Normally, this defaults to "html", but
# can be changed for sites who need it (like for PHP embeded pages).

#HTMLExtension  html

# PageType lets you tell the Webalizer what types of URL's you
# consider a 'page'.  Most people consider html and cgi documents
# as pages, while not images and audio files.  If no types are
# specified, defaults will be used ('htm*', 'cgi' and HTMLExtension
# if different for web logs, 'txt' for ftp logs).

PageType    htm*
PageType    cgi
#PageType   phtml
#PageType   php3
#PageType   pl

# UseHTTPS should be used if the analysis is being run on a
# secure server, and links to urls should use 'https://' instead
# of the default 'http://'.  If you need this, set it to 'yes'.
```

```
# Default is 'no'.  This only changes the behaviour of the 'Top
# URL's' table.

#UseHTTPS      no
```

Figura 5.10 Extracto del archivo webalizer.conf

Antes de generar los reportes es necesario crear el directorio donde serán almacenados, y debe estar ubicado en la ruta que especificamos en el directorio de configuración.

```
$mkdir /var/lib/apache/htdocs/reportes
```

Este directorio debe pertenecer a root y pertenecer al mimo grupo por cuestiones de seguridad, una vez realizado los pasos anteriores el último paso es generar los reportes, con la siguiente instrucción:

```
$webalizer –c /etc/webalizer.conf
```

Para poder ver el resultado de la instrucción anterior solamente necesitamos de un navegador web y escribir la siguiente URL http://www.miservidor.com.mx/reportes , la siguiente imagen muestra la pagina principal de los reportes generados por webalizer.

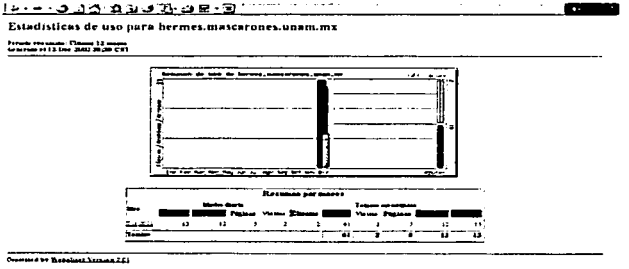


Figura 5.11 Estadísticas de uso durante varios meses.

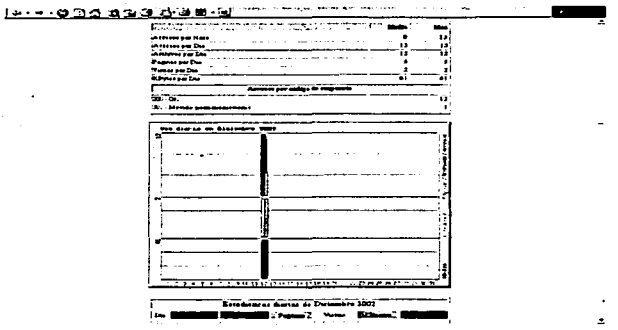


Figura 5.12 Periodo resumido para un mes en especifico.

Como pudimos observar es bastante fácil tener acceso a esta información desde cualquier navegador web solo es necesario saber el nombre del directorio donde se almacenan los reportes, es necesaria agregar algo de seguridad para evitar que alguien fuera de la gente interesada en esta información pueda tener acceso a ella, apache tiene instalado un modulo de seguridad mod_access que nos permite poder proteger un directorio con un login y una contraseña, es necesario agregar algunas lineas en el archivo httpd.conf para que esto pueda ser posible a continuación se muestran las directivas que se deben agregar y cada uno de sus valores si es que lo tiene y su significado.

Por fortuna apache cuenta con esta opción y es posible poner protección por contraseña a un directorio que se puede consultar por Internet, solo hay que agregar algunas lineas de código al archivo de configuración httpd.conf para que funcione.

Enseguida se describen algunas directivas necesarias para poner contraseña a un directorio publico y al final un ejemplo de cómo se implemento en el servidor Hermes.

Las directivas que se utilizan para la seguridad de un directorio publico, estas directivas deben aparecer en una sección **<Difectory>** del archivo de configuración de apache.

Para que solo algunos usuarios permitidos puedan consultar el contenido del directorio publico, es necesarios crear un archivo.htaccess donde se encuentran los nombres de los usuarios que tienen permiso de consultar este directorio, el archivo .htaccess debe estar contenido en el directorio a proteger.

Es necesario activar la directiva AccessFileName que especifica el nombre del archivo de control de acceso que por defecto tiene el nombre .htaccess, hay que notar que se incluye un punto al principio del nombre del archivo de control de acceso esto es para mantenerlo oculto de miradas extrañas.

Esta directiva dentro del archivo de configuración de apache tiene la siguiente forma, por lo regular esta directiva se encuentra descomentada en caso contrario hay que hacerlo.

```
#
# AccessFileName: The name of the file to look for in each #directory
# for access control information.
#
AccessFileName .htaccess
```

Figura 5.13 Archivo de acceso .htaccess

Las directivas que se encuentran dentro de la sección <Directory> se definen a continuación.

La primera directiva es Options esta directiva como su nombre lo indica puede definir varias opciones que a continuación se listan.

ExecCGI
FollowSymLinks
Includes
IncludesNOEXEC
Indexes
MultiViews
SymLinksIfOwnerMatch

La opción **ExecCGI** permite ejecutar programas CGI en el directorio a proteger.

La opción **FollowSymLinks** permite ver desde cualquier navegador los directorios que se definan dentro de la directiva **DocumentRoot** esto quiere decir que todo lo que contenga un directorio puede ser visto y descargado por cualquier persona, inclusive se puede hacer visible otros directorios más sensibles.

La opción **Includes** permite inclusiones del lado del servidor en el directorio de destino, permite el uso del atributo **exec** con lo cual un programador de paginas web puede ejecutar un comando shell.

La opción **IncludesNOEXEC** funciona como la opción includes con la excepción de que el atributo exec no esta permitido.
La opción **Indexes** permite observar un listado del contenido de un directorio, en caso de que en un directorio no haya aun archivo index.html entonces se mostrara todo su contenido.

La opción **MultiViews** permite observar un icono especial para cada tipo de archivo, hasta el momento no hay ningún tipo de problema de seguridad para esta directiva.

La opción SymLinksIfOwnerMatch evita que los usuarios que listan el contenido de un directorio a través de un navegador puedan dirigirse a otro directorio más sensible.

La directiva **AuthName** define el nombre del directorio que sé esta protegiendo, es un comentario que aparecerá en el cuadro de dialogo del nombre de usuario-contraseña.

La directiva **AuthType** define si se utilizan autenticaciones Basic o Digest, la mayoria de los navegadores web solo soportan Basic.

La directiva **AuthUserFile** especifica la ruta del archivo que contiene los nombres de usuarios y las contraseñas.

La directiva **require** es la ultima en la configuración y define los usuarios que se les permitirá acceder.
A continuación se muestra un ejemplo completo que se puede implementar en cualquier servidor apache solo con algunas modificaciones.

```
<Directory "/ruta/reportes">
    Options All
    AuthName "Webalizer Reporte Servidor"
    AuthType Basic
    AuthUserFile /ruta/.htaccess
    require valid-user administrador
</Directory>
```

Figura 5.14 Código para proteger un directorio con contraseña.

La ubicación del archivo htaccess debe ser la misma que la de donde se generan los reportes de webalizer, es decir si se crea un directorio donde se almacenan los reportes de webalizer en ese mismo directorio se guardará el archivo de claves htaccess, A pesar que la clave que se genera en el archivo htaccess esta encriptada existe un script en perl que puede descifrar esta clave así que por seguridad se recomienda cambiar constantemente esta clave y nunca utilizar la misma clave para root y el archivo htaccess.

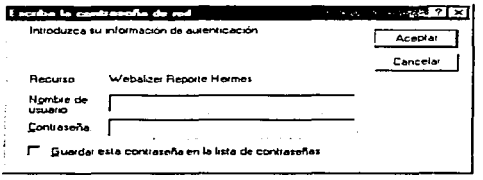
Para crear el archivo .htaccess y poder agregar un usuario valido para la configuración anterior hacemos uso de la instrucción htpasswd, necesitamos ejecutar esta instrucción de la siguiente manera:

Shtpasswd –c /var/lib/apache/htdocs/reportes/.htaccess username

Donde username es el nombre del usuario valido por ejemplo administrador.

Inmediatamente después nos pide un password para este usuario el cual hay que volver a reingresar para confirmarlo. Una vez realizado este proceso ya estamos listo para hacer una prueba a nuestra configuración y ver que funcione correctamente.

A continuación se muestra el resultado que se debe obtener con la configuración anterior



5.15 Cuadro de dialogo del nombre de usuario-contraseña.

Generar reportes y depurar bitácoras cotidianamente resulta un trabajo monoto y bastante pesado a la larga, y más para un administrador que su trabajo no es la administración del servidor web al 100% sino que tiene que distribuir su tiempo en otras actividades, es por eso que se recomienda crear algunos scrpits que nos ayuden en estas tareas podemos hacer uso del demonio crond para programarlas y estas se lleven a cabo por si solas. A continuación se muestra un ejemplo de shell para depurar bitácoras y generara los reportes del servidor.

Un ejemplo de un script para depurar bitácoras del sistema y generar reportes con webalizer.

```
#!/bin/sh
OLD_ACCESS=/directorio/archlog/access.`date +%y%m%d-%H%M%S`
echo El programa para actualizar bitácoras se esta ejecutando
echo Deteniendo apache
/usr/sbin/apachectl stop
Ejecutando webalizer
webalizer –c /etc/webalizer.conf
echo respaldando bitácoras
mv /var/log/access_log `echo $OLD_ACCESS`
echo Creando bitácoras de acceso
```

```
touch /var/log/access_log
echo Arrancando apache
/usr/sbin/apachectl start
echo La actualización termino con éxito
```

Figura 5.16 Script depuralog.sh

Enseguida se muestra otro ejemplo de un script para depurar bitácoras, la diferencia con el anterior es que este segundo ejemplo se ejecuta cada fin de mes.

```
#!/bin/sh
OLD_ERROR=/directorio/archlog/error.`date+%y%m%d%H%M%S`
OLD_MESSAGES=/directorio/archlog/messages.`date +%y%m%d-%H%M%S`
echo El programa para actualizar bitacoras se esta ejecutando
echo Deteniendo apache
/usr/sbin/apachectl stop
echo Respaldando bitacoras error_log y messages
mv /var/log/apache/access_log `echo $OLD_ERROR`
mv /var/log/messages `echo $OLD_MESSAGES`
echo Creando bitácoras de error y mensajes del sistema
touch /var/log/error_log
touch /var/messages
chmod 640 /var/log/messages
echo Reiniciando syslogd
kill -SIGHUP `cat /var/run/syslogd.pid`
echo Arrancando apache
/usr/sbin/apachectl start
echo Termino la actualizacion con éxito
```

Figura 5.17 Script depuralogmen.sh

Para los dos ejemplos anteriores se debe definir el directorio donde estará ubicado el directorio donde se almacenaran los respaldos, el lugar pude ser un home sin privilegios o cualquier otro directorio.

Para programar un cron es necesario saber como trabaja este, la sintaxis de un cron es la siguiente:
Contiene 5 campos que se leen de izquierda a derecha los cuales significan.

- El primer campo representa los minutos (0 - 59).
- El segundo campo representa la hora (0 - 23).
- El tercer campo representa el dia del mes(1 - 31).
- El cuarto campo representa el mes (1 - 12).
- El quinto campo representa el día de la semana (0 – 6) donde 0 significa domingo.

El asterisco para cualquier campo significa todo (días, tiempo).

Ejemplo del programa cron

```
#Trabajo cron que corre a las 12:00 de la noche todos los días para actualización
#de bitácoras
#Trabajo cron de depuración de bitácoras
02 00 * * * /root/depuraacceso
05 00 28 2 * /depraccesomen
05 00 30 * * /depuraaccesomen
```

Figura 5.18 Ejemplo cron.

En el ejemplo anterior ejecutamos cada uno de los script en forma mensual y semanal. La hora y el día dependerán de cada administrador pueden ser ejecutados a una hora y un día determinado cuando no exista demasiada actividad en el servidor.

Programar tareas, respaldar bitácoras y contar con herramientas de detección de intrusos es una actividad sana dentro de la administración de esta manera nos aseguramos que cuando algo extraño ocurra en el sistema contemos con los medios necesarios para obtener información fundamental acerca de lo que esta sucediendo dentro del servidor.

De esta manera protegemos al sistema y a nosotros mismos de posibles problemas en el servidor a futuro, ya sea un usuario que no esta utilizando correctamente su cuenta o tal vez un intento de intrusión al sistema de alguna persona mal intencionada.

CAPÍTULO 6
MANTENIMIENTO

6. Tareas administrativas

Una vez que se configuro el sistema, es necesario encargarse de la administración del sistema por lo que debemos centrarnos en mantener y mejorar tanto el sitio como las herramientas de administración del sistema.

En este capítulo se consideran aspectos importantes acerca de la administración que se realiza en el servidor web del centro.

Algunas de estas tareas son sencillas y rutinarias, mas sin embargo existen algunas tareas que necesitan de mucha paciencia para poder llevarlas acabo como son descargar las actualizaciones de cada paquete que tenemos instalado e instalar estas actualizaciones cuando es necesario, estar actualizado en los reportes de seguridad que se publican continuamente en Internet, solo por mencionar algunas ellas, estas tareas consumen tiempo del administrador y resta atención a otras actividades igual de importantes en su entorno de trabajo. A continuación se mencionan algunas tareas que como administradores son necesarias llevar a cabo continuamente:

- Monitorear la actividad diaria del servidor web.
- Instalación y configuración de nuevo software.
- Actualización del software que ya se encuentra instalado.
- Mejorar las herramientas de seguridad con las que cuenta el sistema.
- Verificar la integridad del sistema.
- Realizar respaldos.

6.1 Monitorear la actividad diaria del servidor.

Una de las tareas importantes de un administrador es saber que actividad esta realizando el servidor web, para esta tarea se utiliza la herramienta webalizer que ya se ha mencionado en el capitulo anterior pero también se puede utilizar las herramientas con las que cuenta el sistema, como los comandos netstat, uptime y top.

El comando netstat es utilizado para monitorear la actividad del servidor, la información que nos despliega en pantalla muestra las conexiones de red establecidas en nuestro sistema, A continuación en la siguiente figura 6.1 se muestra la salida del comando netstat.

```
root@hermes:/home/polo# netstat –an

Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 0.0.0.0:80              0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:25              0.0.0.0:*               LISTEN
tcp        0    912 132.248.75.130:22       200.65.116.237:1801     ESTABLISHED
tcp        0      0 132.248.75.130:80       148.243.82.178:2627     ESTABLISHED
tcp        0      0 132.248.75.130:80       148.243.82.178:2625     ESTABLISHED
tcp        0      0 132.248.75.130:80       200.34.143.22:7258      FIN_WAIT2
Active UNIX domain sockets (servers and established)
Proto RefCnt Flags       Type       State         I-Node Path
unix  7      [ ]         DGRAM                    65     /dev/log
unix  2      [ ACC ]     STREAM     LISTENING     129    /dev/gpmctl
unix  2      [ ACC ]     STREAM     LISTENING     93     /var/run/lprng
unix  2      [ ]         DGRAM                    144409
unix  2      [ ]         DGRAM                    736
unix  2      [ ]         DGRAM                    112
unix  2      [ ]         DGRAM                    106
unix  2      [ ]         DGRAM                    68
```

Figura 6.1 Salida del comando netstat.

La primera parte de la salida netstat **Active Internet connections** muestra las conexiones existentes en diferentes estados y aquellos puertos que están escuchando o esperando conexiones. En el ejemplo anterior podemos observar que la cuarta linea nos muestra una conexión establecida (la columna de estado nos muestra el mensaje **ESTABLISHED**) esta conexión es una sesión ssh, lanzada desde un sistema remoto a nuestro sistema. Esto lo podemos deducir por que en el campo **Local Address** (dirección local) muestra **132.248.75.130:22,** el número tras los dos puntos nos muestra que se trata de una conexión ssh, el segundo campo **Foreing Address** (dirección foránea) nos dice la dirección ip no resulta del equipo remoto y el puerto no privilegiado que le fue asignado.

Todas las lineas cuyo estado es **LISTEN** (escuchando) indica que existe un servicio en el sistema local esperando una conexión remota.

L a segunda parte de la salida de netstat es la sección Active Unix domain sockets, esta parte nos muestra las colas internas y los archivos que se están utilizando para realizar comunicaciones entre procesos.
como netstat es un aplicación que puede ser de mucha utilidad para que el administrador pueda detectar los potenciales problemas de sus sistema, suele ser reemplazada por los hackers cuando toman el control de un sistema, podemos detectar si alguien a modifica el programa netstat si utilizamos alguna herramienta de comprobación de integridad como tripwire o bien podemos utilizar lsof para suplir a

netstat, lsof viene incluido en la mayoria de las distribuciones Linux, pero puede descargarse de ftp://vic.cc.purdue.edu/pub/tools/unix/lsof.

La herramienta lsof utilizando la opción –i nos muestra los puertos de red que están escuchando y el programa que lo esta ejecutando. En la siguiente figura 6.2 podemos observar una salida del comando lsof.

```
root@hermes:/home/polo# lsof -i
COMMAND   PID  USER  FD  TYPE DEVICE SIZE NODE NAME
sshd       84  root   3u IPv4   79      TCP *:ssh (LISTEN)
sendmail  100  root   4u IPv4  107      TCP *:smtp (LISTEN)
httpd   20703  root  16u IPv4 60067      TCP *:http (LISTEN)
httpd   23394 nobody  16u IPv4 60067      TCP *:http (LISTEN)
httpd   23395 nobody  16u IPv4 60067      TCP *:http (LISTEN)
httpd   23396 nobody  16u IPv4 60067      TCP *:http (LISTEN)
httpd   23397 nobody  16u IPv4 60067      TCP *:http (LISTEN)
sshd       84  root   3u IPv4   79      TCP *:ssh (LISTEN)
sendmail  100  root   4u IPv4  107      TCP *:smtp (LISTEN)
sshd     23887  root   4u IPv4 169378       TCP hermes.mascarones.unam.mx:ssh-
>dup-200-65-116-237.prodigy.net.mx:1801 (ESTABLISHED)
```

Figura 6.2 Salida del comando lsof.

Como podemos ver en el ejemplo anterior algunos servicios son ejecutados por inetd y otros por programas independientes, vemos mas a la derecha en el campo **NAME** (nombre) es sustituido por el nombre del puerto estos datos son obtenidos del archivo /etc/services.

El siguiente comando que analizaremos es el comando uptime, este aplicación muestra información menos detallada que netsat y solo indica el tiempo que el servidor lleva trabajando, él numero de usuarios conectados, cuantos procesos se están ejecutando y la cantidad de memoria que están consumiendo.

A continuación se muestra en la siguiente figure 6.3 un ejemplo del comando uptime

```
polo@hermes:~$ uptime

 4:25pm  up 81 days, 20:35,  1 user,  load average: 0.00, 0.00, 0.00
```

Figura 6.3 Salida comando uptime

Otra herramienta que nos auxilia en el monitoreo del sistema es la herramienta top, que viene incluida solo en distribuciones Linux, esta herramienta nos muestra información de los procesos que se están ejecutando en el sistema a diferencia de la herramienta uptime la información que top nos muestra se actualiza constantemente.

En la siguiente figura 6.4 tenemos un ejemplo de la utilidad top.

```
13:09:46 up 64 days, 18:45,  1 user,  load average: 0.00, 0.00, 0.00
47 processes: 46 sleeping, 1 running, 0 zombie, 0 stopped
CPU states:  0.2% user,  0.0% system,  0.0% nice,  99.8% idle
Mem:   126108K total,   121820K used,    4288K free,    13740K buffers
Swap:  136544K total,     2784K used,   133760K free,    33752K cached

  PID USER     PRI  NI  SIZE  RSS SHARE STAT %CPU %MEM   TIME COMMAND
23984 root      14   0   968  968   768 R     0.1  0.7   0:00 top
    1 root       8   0    76   64    48 S     0.0  0.0   0:04 init
    2 root       9   0     0    0     0 SW    0.0  0.0   0:00 keventd
    3 root      19  19     0    0     0 SWN   0.0  0.0   0:00 ksoftirqd_CPU0
    4 root       9   0     0    0     0 SW    0.0  0.0   0:11 kswapd
    5 root       9   0     0    0     0 SW    0.0  0.0   0:00 bdflush
    6 root       9   0     0    0     0 SW    0.0  0.0   0:00 kupdated
    7 root      -1 -20     0    0     0 SW<   0.0  0.0   0:00 mdrecoveryd
   76 root       9   0   424  376   340 S     0.0  0.2   0:00 syslogd
   79 root       9   0   248  180   180 S     0.0  0.1   0:00 klogd
   81 root       8   0   312  240   224 S     0.0  0.1   0:00 inetd
   84 root      13   0   464  328   292 S     0.0  0.2   0:10 sshd
   95 root       6   0   348  328   268 S     0.0  0.2   0:00 crond
   97 daemon     9   0   356  292   276 S     0.0  0.2   0:00 atd
  100 root       9   0   768  560   448 S     0.0  0.4   0:00 sendmail
  103 smmsp      9   0   708  448   368 S     0.0  0.3   0:00 sendmail
  112 root       9   0   240  184   184 S     0.0  0.1   0:02 gpm
20703 root      11   0  1548 1516  1456 S     0.0  1.2   0:00 httpd
```

Figura 6.4 salida de top.

En ejemplo anterior vemos la salida del comando top, esta herramienta maneja una gama completa de opciones que podemos ejecutar directamente desde la interfaz de top.

El monitoreo del sistema debe ser frecuente, se debe ver los reportes generados por webalizer, además de leer los mensajes que genera el sistema todo esto para saber si su funcionamiento es correcto y en caso de algún problema este sea detectado a tiempo.

6.2 Instalación y configuración de nuevo software.

En ocasiones por necesidades del mismo centro es necesario instalar software con el que no cuenta la distribución que se tiene instalada en el servidor y es necesario descargar e instalar este software para que sea utilizado por los usuarios que lo requieran.

En ocasiones el software que se descarga necesita de otras bibliotecas para trabajar, que también hay que descargar e instalar, este es un trabajo tedioso pero interesante, se aprende mucho acerca de algunas características del software que se instala cuando se hace en Linux , ya que en Windows solo se dan algunos botonazos de ratón y solo el programa sabe donde instalo cada cosa, en Linux esto no sucede, ya que se puede personalizar la instalación tanto como se desee manteniendo el control total durante este proceso, sin embargo el rendimiento, control, desempeño y seguridad es muy superior en UNÍX y Linux comparado a Windows.

Este capitulo es una recopilación de consejos acerca de que hay que hacer para instalar un paquete en Linux, esto parecerá obvio para aquellos que han instalado en mas de una ocasión un paquete en Linux pero para las personas que nunca lo han hecho es una tarea bastante compleja y más si son personas que nunca han trabajado en Linux o que vienen de una plataforma distinta como Windows.

Lo primero que se debe hacer antes de instalar un paquete en Linux es encontrar el lugar en donde se puede descargar una copia del software que sé esta buscando, se puede hacer directamente en un buscador o intentando encontrar la pagina principal del producto.

Se recomienda descargar la versión estable del paquete, de esta forma sé esta seguro que los errores de otras versiones están corregidos, por lo regular los paquetes que son para plataformas UNIX vienen con una extensión.tar.gz, una vez que se descarga el paquete hay que quitar esta extensión con la ayuda de los comandos tar y gzip. Un ejemplo de cómo se aplican estos comandos en conjunción seria de la siguiente manera:

Sgzip -dx archivo.tar.gz | tar xvf –

o bien

Star zxvf archivo.tar.gz

De esta forma se genera un directorio con el nombre del paquete, el siguiente paso es entrar en directorio y editar el archivo README o INSTALL con cualquier editor de textos como pico o vi, aquí se encontrara toda la documentación necesaria para saber los pasos a seguir para la instalación del paquete y si hace falta descargar otro paquetes que sean necesarios para la instalación del software, hay que tomar en cuenta que la mayoría de las veces estos archivos estarán escritos en el idioma ingles.

Si no es suficiente la documentación que aquí se encuentra se puede consultar la pagina del fabricante e investigar si tiene un manual en nuestro idioma o una mas información de la que viene incluida en el archivo README.

Una vez que finaliza la instalación el paquete instala varios archivos en diferentes rutas dentro del árbol de directorios de Linux, por lo regular siempre existe un archivo binario que ejecuta el programa que se acaba de instalar y otro que guarda la configuración del

programa el cual puede ser editado y modificado de acuerdo con las necesidades que se tengan o simplemente se deja con los valores por omisión sin modificarlo.

Solo hay que leer la documentación completa del paquete para saber cual es el nombre de estos archivos, en el caso de archivos binarios se recomienda escribir el nombre del paquete en la línea de comandos una vez que se instalo y después oprimir la tecla tab varias veces lo cual debería dar como resultado si es que lo hay el nombre de varios programas que están instalados en Linux se debe verificar si entre ellos esta el que se necesita, de esta forma se sabe que el paquete ha sido instalado con éxito.

En general estos son los pasos para instalar cualquier paquete, la forma de trabajar con él dependerá del administrador del sistema.

6.3 Mejorar herramientas de seguridad.

Una de las tareas principales que tiene un administrador es la de encontrar herramientas de seguridad lo bastante robustas para la protección del servidor, existen herramientas bastante buenas pero que no son software libre y que por el contrario se debe pagar una cierta cantidad por su uso, esto no seria un serio problema si se contara con los recursos suficientes para comprar una copia de este software y poder probar todas sus características en el sistema con el que se trabaja.

En ocasiones algunos de estos productos regalan pruebas de su software pero solo por 30 dias, tiempo suficiente para probar algunas de sus caracteristicas pero finalizado este periodo el software no corre o tan solo corre con algunas características, entonces no-queda mas buscar herramientas donde no se tenga que pagar nada por su uso, que sean de uso libre para cualquier tipo de propósito ya sea educacional, personal o comercial, existen en el mundo del software libre muchas herramientas que funcionan mejor que algunas herramientas comerciales, solo hay que buscar en el lugar indicado.

Existen sitios en Internet donde se puede descargar una gran variedad de herramientas para su uso en Linux, algunos de estos sitios ofrecen una pequeña descripción acerca de las caracteristicas del programa que puede ser de gran ayuda cuando se busca algo en concreto.

Algunos sitios donde se puede encontrar y descargar software gratuitamente son:

http://www.freshmeat.net
http://www.tucows.com

Algunas de las mejores herramientas de seguridad que se han inventado están disponibles en la mayoría de las distribuciones Linux, y otras que no se incluyen hasta versiones mas recientes, pero se pueden descargar y compilar sin importar la distribución ni el kernel, en algunas ocasiones. En otras solo será suficiente descargar algunos archivos extras para que funcione correctamente.

Una buena referencia para aprender acerca de herramientas de seguridad y en español es la pagina del departamento de seguridad de la UNAM, en este sitio se encuentran avisos de los últimos sucesos que han ocurrido con respecto a seguridad se incluyen algunos tutórales que pueden servir como ayuda para comenzar a conocer las diferentes herramientas de seguridad que existen y como se emplean.

http://www.seguridad.unam.mx

Otros lugares que se pueden consultar son los sitios del CERT y securityfocus donde se puede encontrar avisos acerca de la seguridad de los diferentes SO incluyendo Linux, además de que cuenta con manuales para la instalación y manejo de software de seguridad un inconveniente de estas dos paginas para algunas personas es que están escritas en el idioma ingles.

http://www.cert.org
http://www.securityfocus.com

6.4 Respaldos.

Los respaldos son un tema importante en la administración, no puede existir un sistema sin respaldos, si esto ocurre se arriesga a no poder recuperar el sistema en forma completa o parcial en caso de un daño grave.

El administrador debe adquirir la cultura del respaldo de datos en un host UNIX, esto puede ayudar a solucionar problemas de diversos tipos pueden ocasionar la perdida de datos, como la eliminación accidental de archivos, fallas de hardware, etc.

El administrador debe tomar en cuenta los beneficios de los respaldos, debe planificar sus respaldos, no tan solo es realizar un respaldo y nada mas, sino que se debe estructurar perfectamente los pasos que se debe seguir por ejemplo:

- Los respaldos serán completos o increméntales.
- Conocer las herramientas con las que cuenta el sistema para realizar respaldos.
- Saber con que medios se cuenta para realizar los respaldos (unidades de cinta, otra partición, quemadores, etc.)
- La documentación de los respaldos.

6.4.1 Respaldos completos o increméntales.

El respaldo completo copia todos los archivos, pero es necesario responder a la pregunta: ¿es necesario realizarlo todos los dias?. Un respaldo completo requiere. Por lo general, gran cantidad de tiempo y suficiente medio de respaldo que pueda guardar todos los archivos del sistema.

Un respaldo incremental copia los archivos que han cambiado desde él ultimo respaldo completo.

Los sistemas de archivos activos se deben respaldar con cierta regularidad; mientras que otros archivos se pueden respaldar con menor frecuencia.

Debemos asegurarnos de tener copias de todos los sistemas de archivos y de que estén actualizados.

En un servidor web los datos no cambian con tanta frecuencia así es que en el caso de Hermes los respaldos se realizan en forma incremental, solo respaldan aquellos datos que han sido modificados.

6.4.2 Herramientas de respaldo en Linux.

Existen algunos comandos relativamente simples que se han utilizado desde hace mucho tiempo en sistemas UNIX para la creación de respaldos, como tar y cpio.

Puesto que el respaldo y la restauración de archivos son aspectos muy importantes, hay varios comandos disponibles dedicados a esa tarea, como ejemplo, tar que es una herramienta para respaldar en cinta, disponible en todo sistema UNIX y cpio que es una herramienta de propósito general para el respaldo de archivos, también esta disponible en todos los sistemas UNIX.

A continuación se muestran algunas características de estos dos comandos.

Tar: programa que archiva, diseñado para almacenar y para extraer archivos de un archivo conocido como archivo tar. Se puede realizar copias de seguridad en cintas y dispositivos.
Sintaxis:

Star [opciones] archivo tar o dispositivo

opciones:

-c crea un archivo tar.
-t lista los nombres de archivos que hay en un archivo tar.
-r añade archivos a un archivo empaquetado
-x extrae archivos de un archivo empaquetado.
-f nombre-archivo-empaquetado cuando se da un nombre-archivo empaquetado, la opción -f guarda el archivo empaquetado tar en un archivo con ese nombre.
-v visualiza cada nombre de archivo a medida que lo va empaquetando.

El comando cpio copia archivos en un archivo. Lee una lista de nombres de archivo, una por linea, en la entrada estándar, y escribe el archivo sobre la salida estándar.

El comando cpio tiene dos modos de operación:

El primero utiliza la opción -o para copiar archivos en un archivo y la otra, usando la opción -i, para extraer archivos de un archivo. El archivo puede ser otro archivo en el disco, una cinta magnética, etcétera.

En primer lugar se necesitan generar la lista de nombres de archivos usando una orden tal como ls o find.

Sintaxis:

$nombre-generado | cpio -o > archivo o dispositivo

$cpio -i nombres-de-archivos < archivo o dispositio
-o mediante esta poción, significa "out", cpio crea una salida almacenada que puede ser redireccionada hacia un archivo, convirtiéndolo en un archivo respaldo.

Ejemplo:

$ ls | cpio -o > mihome

$ find -name * | cpio -o > mihome

Mediante la opción -i, que significa "in", cpio extrae archivos de un archivo el archivo se lee desde la entrada estándar, que ha sido redireccionada desde un archivo respaldo.

ejemplos:

$cpio -i < mihome

Estas dos herramientas solo crean un archivo que generalmente es bastante grande en este caso es necesario que este archivo sea más pequeño para que no ocupe mas espacio que el necesario en este caso se utiliza el comando gzip, esta herramienta comprime un archivo, a continuación se muestran algunas características de esta herramienta.

La utilidad gzip es la herramienta estándar de compresión que se utiliza en la mayoría de los ambientes UNIX, además de comprimir y también descomprime archivos.

Sintaxis:

$gzip [opciones] nombre-de-archivo

El comando gzip reemplaza el archivo con una versión comprimida del mismo archivo que tendrá el mismo nombre y la extensión gz

opciones:

-e envia la versión comprimida del archivo a la salida estándar.

-d descomprime un archivo comprimido.
-h visualiza un listado de ayuda.
-l visualiza el tamaño comprimido y sin comprimir de cada archivo de la lista.

Ejemplos:

Sgzip -l archivo.gz
Sgzip -d archivo.gz
Sgunzip archivo.gz

Con la ayuda de las herramientas anteriores podemos crear respaldos de todo el sistema si es necesario o solo la información que sea importante para nosotros..

6.3 Medios de respaldo.

Dependiendo de los sistemas instalados en el sistema, se pueden utilizar cintas de 9 pistas, cartuchos de cinta de 1/4 de pulgada, cinta DAT de 4 u 8 milímetros, discos flexibles o quemadores de CD-ROM.

Cada uno tiene sus propias ventajas y/o desventajas en términos de espacio físico que ocupan, la cantidad de información que pueden guardar y el costo de los dispositivos y el medio.

En el caso de Hermes se cuenta con una partición separada donde se almacenan estos respaldos, de esta manera no se ocupa espacio necesario para otras aplicaciones, en caso de un daño grave se tienen respaldos de estos archivos en otros servidores y en dispositivos de almacenamiento como discos duros y cdrom.

6.4 Documentación de los respaldos.

Es importante documentar todo lo que se realiza en el servidor esto es útil en caso de que el administrador actual se vaya y venga otra persona a tomar su lugar, de estar manera para el siguiente administrador será mucho más fácil aprender todo lo necesario del sistema y su adaptación será mucha más rápida.

Se deben documentar cosas como:

- Cada cuando se debe realizar los respaldos.
- Como se realiza un respaldo.
- La forma como se deben etiquetar los respaldos.
- Donde se deben guardar los respaldos.

El tamaño de cada uno para ver su crecimiento y prever las actualizaciones necesarias.
Es importante contar siempre con los registros documentados y sustentados debidamente que permitan a otros administradores apoyar o continuar con el trabajo que se sustentan en el servidor.

CONCLUSIONES.

A manera de conclusiones se puede establecer lo siguiente:

Linux es un sistema robusto, estable y confiable, una alternativa excelente si se busca proporcionar servicios de Internet de algún tipo.

Gracias a su rápido desarrollo podemos contar con las herramientas necesarias para tener con un sistema mas seguro, fácil de administrar y actualizar.

Toda la información es publica y no es propiedad de algunos cuantos, existe una comunidad que se encarga que esto sea así durante mucho tiempo.

En un país como el nuestro contar con tecnología de este tipo a disposición de cualquier persona, permite el desarrollo de empleos y el desarrollo de profesionistas, profesores, estudiantes, etcétera. Para poder innovar y no tan solo consumir tecnología.

En un tiempo no muy lejano podremos ver empresas basadas 100% en linux, compañías proporcionando soporte a usuarios y empresas que estarán interesados en esta tecnología, profesionales vendiendo sus servicios para soluciones linux, gente que se encargué de enseñar este sistema operativo, lo cual significa para países del tercer mundo como el nuestro una alternativo para poderse desarrollar sin tener que invertir grandes cantidad de dinero en adquirir licencias de uso de software.

En México linux se utiliza en el Departamento del Distrito Federal, la Asamblea Legislativa del Distrito Federal está preparando una propuesta para institucionalizar en todas las instancias gubernamentales locales a utilizar software libre en vez de propietario. En caso de aprobarse, entraría en vigor a finales de este año (2003) o inicios del siguiente (2004). Tampoco se descarta, según la nota, que se extienda al gobierno federal.

Otros países en el mundo como Alemania, España y EUA, han echo lo mismo y su uso ha sido exitoso.

A nivel mundial el uso de software libre sin marca y hecho a la medida cobra cada vez mayor fuerza, sobre todo en ambientes gubernamentales y empresariales.

El estandarte de este mercado es Linux, al que se le ha considerado un rival de Windows. Linux sube su presencia entre 28 por ciento y 35 por ciento cada año y porque las soluciones basadas en él son hasta 90 por ciento más baratas.

Se espera que para 2004 Linux obtenga en ciertos nichos hasta 50 por ciento del mercado a nivel mundial, gracias al apoyo recibido por marcas como IBM, HP y el súbito incremento de *fans* alrededor del mundo

China y Brasil son los países que más lo utilizan. Los gobiernos de Francia, Alemania y China están totalmente bajo Linux, Amazon ha ahorrado 32 millones de dólares al montar todo su sitio en Linux, Trend Micro 30 millones e Intel, casi 200.

Por todas estas características en muy poco tiempo linux será una alternativa fuerte para poder implementar toda una red de servicios por lo cual debemos estar preparados para cuando esto ocurra.

BIBLIOGRAFIA

Servidor Apache al descubierto.
Rich Bowen.
Ken Coar.
Prentice Hall.

Linux guía de instalación y administración..
Vicente López Camacho.
Ignacio García Soler.
McGraw-Hill.

Manual de administración de Linux..
Steve Shah.
McGraw-Hill.

Linux máxima seguridad.
Anónimo.
Prentice may.

Linux manual de referencia.
Richard Petersen.
McGraw-Hill.

Linux serie práctica.
M. Drew Streib.
Michael Turner.
Prentice Hall.

Linux edición especial.
Jack Tackett.
David Gunter.
Prentice Hall.

Red Hat Linux 6.
Arman Danesh.
Anaya.

Red Hat Linux 6 a fondo.
David Pitts.
Bill Ball.
Anaya.

Guía avanzada de administración de sistemas Linux.
Carling M.
Prentice Hall.

PÁGINAS WEB

"Debian Linux Installation & Getting Started" se puede encontrar en http://www.linuxgazette.com/issue15/debian.html.

"Linux installation & Getting Started" se centra principalmente en slackware, se puede encontrar en http://durak.org/sean/pubs/ligs-slackware/node1.html.

Pagina principal del proyecto apache.
http://httpd.apache.org/

Pagina principal de Linux Slackware.
www.slackware.com

Revista electrónica Linuxfocus
http://tldp.org/linuxfocus/