

Curso practico de Linux



- [Notas del autor](#)
- [Un poco de historia](#)
 - [UNIX](#)
 - [Linux](#)
- [Instalando Linux](#)
 - [Booteando con Loadlin](#)
- [Comandos basicos](#)
- [Sistema de archivos](#)
 - [Administracion de usuarios](#)
 - [Agregar un usuario](#)
 - [Eliminar un usuario](#)
 - [El archivo passwd](#)
 - [Manejo de sesiones](#)
- [Sistema de permisos](#)
 - [Forma Absoluta](#)
 - [Forma Relativa](#)
- [El kernel](#)
 - [Configuracion](#)
 - [Compilacion del nucleo](#)
- [Control de tareas](#)
 - [Ejecucion en primer y segundo plano](#)
 - [Envio de procesos a primer y segundo plano](#)
 - [Detencion de procesos](#)
- [Links de interes](#)

(c) 2002, Mariano Pancaldi, [Sol-Tech \[SH\]](#)

Curso practico de Linux



Ultima Revision:
07/08/2002.

Copyright (c) [Mariano Pancaldi](#), 2002. Elaborado a partir de experiencia propia y de información recopilada a traves de Internet. Vaya mi agradecimiento para Facundo Gimenez por organizar este caos de informacion.

Notas del autor.

Debo dejar en claro que este documento es totalmente gratuito, se puede distriibuir y publicar en cualquier sitio web, pero no comercializar.



Queda terminantemente prohibida su venta sea Online, en Disquettes o en CDs

El fin de este curso es orientar a usuarios que no conocen aún o recién se inician en el fascinante mundo de Linux. Si usted quiere contactarse conmigo para enviarme algun comentario o algun otro tema que no haya desarrollado para agregar a este documento por favor envieme un mail a curso@sol-tech.com.ar

Al desarrollar este curso totalmente bajo Linux, pido disculpas por mis errores gramaticales y/o de expresión. Debido a que no he instalado en mi sistema ningun procesador de textos con funciones de corrección gramatical en Castellano, prometo corregir este error en futuras versiones de este curso :-)

powered by



HAGA CLIC AQUI PARA BAJAR EL CURSO EN FORMATO ZIP: [Descargar Curso](#)

[Indice General](#)

Un poco de historia

Transcurre el tiempo y mas gente sabe que existe, suma mas adeptos, ya no es el sistema operativo exclusivo de los gurus de la informática sino que se ha volcado a los usuarios estandares. Tal vez usted sea uno de ellos, que tiene ganas de conocer mas por dentro este fabuloso sistema operativo que da ganas de seguir contribuyendo.

UNIX, El padre de la criatura.

UNIX es un sistema operativo multiusuario de uso general desarrollado por Bell Telephone Laboratories a fines de la decada de los 60 por miembros del personal técnico. Este laboratorio no es solo el laboratorio industrial de investigación mas grande del mundo sino también el de mas prestigio científico.

Unix comenzó como un proyecto "informal" a espaldas de la direccion de Bell Labs. Una de las principales motivaciones de los autores, Ritchie y Thompson, fue su insatisfacción con los sistemas operativos existentes. Sus ideas eran crear un sistema operativo "flexible" de ahí la característica esencial de Unix: Una estructura de archivos jerárquica en forma de arbol lo cual provee un medio lógico y sencillo para almacenar en forma transparente archivos de distintos tipos y distintas aplicaciones.

¿POR QUE UNIX ?

Porque es un sistema operativo estandard que permite que el mismo software se ejecute en computadoras de esta portabilidad no está ligada al diseño del hardware de una computadora, como es el caso de los sistemas operativos escritos en lenguaje ensamblador, puede instalarse en cualquier computadora para el que se haya escrito un compilador de lenguaje C, con el que fue diseñado. Esto implica que puede correr en procesadores como el Intel 8080, Zilog 80, Intel 8086, 80286, 80386, Motorola 68000, PDP-11, VAX, HEWLETT-PACKARD 9000, IBM 370 Y AMDAHL. Además Unix ofrece utilidades para comunicación y permite la transferencia de aplicaciones entre diferentes maquinas.

DE LAS 13.000 LINEAS DE CODIGO FUENTE DEL SISTEMA, SOLO 800 SON EN ASSEMBLER; ESTO LO HACE PORTABLE

1990, El año que hicimos contacto.

Fuera de todo monopolio y gracias a miles de programadores de todo el mundo, se lo puede conseguir desde el Ingles hasta el Aleman, del Español al Danes, y esto es posible gracias a la idea de un joven Finlandes de nombre Linus Torvalds quien por finales de 1990 y finalizando su carrera de Informática en la universidad de Helsinki, Finlandia, se le ocurrio desarrollar una version hogareña del sistema operativo Minix (popular entre los jovenes programadores en aquellos tiempos, Minix era el Clon gratuito de UNIX). Todo partió desde un archivo "main.c", pocos meses transcurridos Linus ya tenia desarrollado un hibrido con el cual se podian realizar

algunas operaciones básicas como por ejemplo copiar archivos y montar unidades. Por suerte para él y el apoyo recibido en la universidad pudo continuar con el desarrollo de su propio sistema operativo. Un día recorriendo el jardín zoológico de Helsinki se acercó a la jaula de los pingüinos, animales que le llamaban la atención y al pasar su mano entre los barrotes uno de los pingüinos se aferro a uno de los dedos de su mano. Esto le provoco una infección que desencadenó en un estado febril muy importante.

Una vez recuperado se le ocurrió utilizar como logo a este animal y lo denominó Tux, ya que según el sería lo más anticomercial posible. Entonces las piezas empezaban a encajar en el rompecabezas, tenía el código fuente en proceso, y poseía un logo. Pero ¿Cómo podría finalizar un sistema operativo tan grande? y la respuesta fue sencilla, comenzó a contactar a jóvenes de todo el mundo por internet (la mayoría de ellos estudiantes de informática en otras universidades que se comunicaban por bbs locales o por medio de internet, con conocimientos elevados "hackers". Como se los conoce normalmente). Lanzando un manifiesto que convenció a gran cantidad de ellos el sistema operativo se desarrolló rápidamente y para contribuir con la comunidad a modo de reveldía el sistema se desarrolló bajo las normas del "Free Code" (Código Libre). Linux consiste en el sistema operativo en sí llamado "kernel" más una serie de programas asociados. El kernel, como todos los tipos de Unix, es un sistema operativo multitarea y multiusuario. Aunque usa una estructura de archivos diferente, y de ahí que no sea directamente compatible con DOS y Windows, es tan flexible que muchos programas DOS y Windows podrán ejecutarse mientras estemos en Linux, o sea que un usuario probablemente querrá arrancar en Linux y después ser capaz de correr programas DOS y Windows desde Linux.

Programas asociados que vienen con la mayoría de distribuciones de Linux pueden incluir:

- Un programa "shell" (Bourne Again Shell, la shell bash es la más común)
- Compiladores para lenguajes de programación tales como C (propio de los sistemas UNIX), C++, ADA, PASCAL, PERL, MODULA 2, FORTRAN, PYTHON, EIFFEL, BASIC, RPG, SMALLTALK, COBOL, ASSEMBLER, OBERON.
- X "X Window", una interfaz de usuario gráfico
- Programas de utilidades como el lector de email Pine, Elm, Midnight Commander, Sendmail, etc.

Debo aclarar que la comunidad que desarrolla actualmente un Linux totalmente gratuito es la comunidad GNU/Linux Debian. Linux tiene copyright bajo la GNU General Public License. Bajo este acuerdo, Linux puede ser redistribuido a cualquiera junto con el código fuente. Cualquiera puede vender una variante de Linux, modificarla y volverla a embalar. Pero incluso si alguien modifica el código fuente él o ella no exigirán copyright por nada creado a partir de Linux. Cualquiera que venda una versión modificada de Linux deberá proveer el código fuente a los compradores y permitirles usarlo en sus productos comerciales sin cobrar cuotas de licencia. Este acuerdo se conoce como "copyleft". Existen muchas versiones de Linux en caja o comerciales.

Linux Red Hat, creada por Red Hat Software, en Connecticut, EE.UU. Una de las

distribuciones más fáciles de utilizar el cómodo mantenimiento de componentes de software, lo que facilita enormemente las tan frecuentes actualizaciones. Se puede obtener tanto gratuitamente en la red como adquiriendo el CDROM en caja correspondiente. Otras empresas comercializan también sistemas basados en Red Hat, como Caldera Inc., Pacific Hi-Tech y Conectiva (Linux desarrollado por programadores Argentinos y Brasileños). El sistema se basa bajo la licencia de GNU y es gratuito a los demás desarrolladores, por lo que es previsible que en el futuro otros muchos asuman este sistema en sus propias distribuciones, lo que facilitará enormemente las actualizaciones.

Linux Slackware, esta distribución es quizás de las más extendidas en todo el mundo y la más conocida. Incluye con creces todo el software que cualquier usuario medio o avanzado pueda necesitar, y su método de instalación es bueno aunque lejos de las típicas y vistosas instalaciones a las que el usuario está acostumbrado en los sistemas comerciales. Tiene diversos derivados comercializados bajo diferentes nombres (por ej. "Slackware Pro") que mejoran la instalación e incluyen manuales y bibliografía.

Linux Debian, la Free Software Foundation (FSF) es bien conocida entre los usuarios de software gratuito para Unix. Son los creadores del sistema GNU, su futuro Unix gratuito. Ya hay mucho material pero no un sistema operativo completo, así que mientras tanto ofrecen un Unix integrado por el núcleo de Linux y el software de GNU.

Eceptuando Linux Debian el resto de las distribuciones pueden bajarse gratuitamente pero solo los usuarios que adquieren el producto en caja tienen acceso al soporte técnico de cada compañía.



Algunas razones para instalar o considerar a Linux como su próximo sistema operativo:

- Es uno de los sistemas operativos más actualizados y utilizados, se calcula que el 75% de los servidores que existen actualmente en Internet corren bajo sistemas Unix o Linux.
- Es muy, muy seguro y no complejo de configurar, es casi imposible que un virus infecte su sistema si usa Linux, ya que solo existen unos pocos que ataquen al mismo.
- Miles de programadores en todo el mundo desarrollan aplicaciones todos los días, algunas con soporte multilingüe.
- Es 100% Multitarea y no requiere un equipo muy elevado para poder ejecutarse.
- No importa que versión de Linux instale todas las aplicaciones existentes corren en la mayoría de las versiones Linux.
- Navegar por la red le resultará mucho más atractivo, cuenta con Netscape Communicator, Mozilla y muchos browsers más.
- Cuenta con documentación detallada de la mayoría de los programas del sistema, además de los HOWTO's que son documentos de ayuda mucho más detallados.
- Uno de los puntos muy fuertes de Linux es su gran conectividad. Incluso con recursos

modestos en hardware puede ser un hábil servidor de red y convivir prácticamente con cualquier configuración que ya tengamos funcionando en nuestra red. A medida que mejoremos el equipamiento, Linux aprovechara todos los recursos convirtiéndose en un potente centro de servicios.

- Es Multilenguaje, se puede instalar en el idioma que uno desee.
- Todos los procesos tienen garantizada una zona protegida de memoria para su ejecución, sin que el mal funcionamiento de una aplicación cuelgue a todo el equipo.
- Para conectarse a Internet por ejemplo en Windows necesitamos instalar el protocolo TCP/IP. Esto no ocurre en Linux ya que al ser TCP un protocolo propio de los sistemas UNIX luego heredado por Linux contamos directamente con él en el kernel, incluso la aparición de Java no paso inadvertida para Linux de manera que fue inmediatamente absorbida y ya se cuenta con soporte tanto para JavaScripts como para applets Java los cuales se pueden ejecutar como software nativo al estar incluido en el Kernel.

[Indice General](#)

[Glosario de terminos](#)

Instalando Linux

Si bien el sistema esta diseñado para correr en equipos i386. El tipo de arquitectura que nuestra computadora necesitará para poder instalar un sistema Linux es variado, dependiendo de las exigencias que el usuario necesite. Partiendo como equipo 486/66Mhz con 8 Megabytes de RAM yo no he tenido problemas, unos 512 Mb de espacio en disco. Aunque se recomienda un procesador Pentium o AMD de 166Mhz o superior ya que con el trascurso de los años el sistema fue creciendo sobre todo en el modo gráfico, es mas exigente, dependiendo del entorno que usemos. Algunas alternativas de instalacion para diferentes arquitecturas de equipos son:

Procesador 486

Si el sistema se utilizara como cliente es recomendable personalizar la instalación, Si se instala el servidor X Window utilizar un servidor de entorno pequeño como el blackbox o el enlightenment que si bien no traen muchos recursos preinstalados son totalmente configurables. Estos equipos son recomendables para diseñar firewalls (barreras de fuego), Routers (Pasarelas) y Proxys.

Procesador Pentium

Soporta la instalacion estandard y modo gráfico con servidores como el KDE o el GNOME., si utilizamos el editor de textos StarOffice deberémos tener un poco de paciencia al momento de la carga.

Procesador Pentium II

Soporta todos los modulos sin problemas y los servidores KDE2 y GNOME. Ejecuta sin lentitud el editor de textos StarOffice. Es la arquitectura ideal para utilizar el sistema como servidor o cliente.

Para finalizar estos son algunos de los requisitos que Linux necesita para poder instalarse:

Bus de E/S: Su computadora debe utilizar el bus ISA, EISA, PCI, o VL. El bus VL se conoce tambien como VESA Local Bus o VLB. Las computadoras que tienen PCI o VLB generalmente tienen también ranuras ISA o EISA.

RAM y Disco Rigido: Debe tener al menos 4Mb de RAM aunque es recomendable no menos de 16Mb y 150Mb de disco duro, pero si usted quiere instalarlo todo necesitará 500Mb o más. Las interfaces de disco que emulan la interfaz de disco "AT" que se llaman habitualmente MFM, RLL, IDE, o ATA están soportados. Las controladoras de disco SCSI de diversos fabricantes estan soportadas.

MPC: Como casi todas las distribuciones de Linux vienen en CD-ROM necesitaremos una lectora que puede ser IDE, SCSI o con norma propia como alguna Creative o Mitusmi.

Placa de video: Aunque soporta desde una hercules, es recomendable usar una placa de video compatible con VGA para la terminal de la consola. Prácticamente cualquier tarjeta gráfica

moderna es compatible con VGA. CGA, MDA, o HGA pueden funcionar correctamente para texto, pero no funcionar con el X Window.

Otro Hardware: Linux soporta una gran variedad de dispositivos como son: mouses, impresoras, scanners, modems, placas de red, etc. Sin embargo, no se requiere ninguno de estos dispositivos durante la instalacion del sistema.

Comenzando la instalacion

Intentaré hacer lo mas sencillo posible la este capítulo, primero tendra que plantearse lo siguiente. Si desea hacer convivir a Linux con otro sistema operativo en el mismo disco, por ejemplo, si deseo hacer convivir a Linux con Windows en el mismo disco, o bien, instalar el sistema en otra unidad de disco totalmente aislado de otro sistema.

Antes de proceder a la instalacion en si, debemos realizar una serie de preparativos en nuestro equipo.

Preparacion del disco rigido:

Si no ha particionado aún su rígido para Linux y memoria virtual deber recurrir a la aplicación shareware FIPS que encontrará en cualquier distribución de Linux, pero antes de usarla asegurese de que desfragmentó su HD. FIPS le permitir ahora particionar su rígido sin que pierda la información contenida.

Dejaremos de esta manera suficientes megas libres de su disco rígido para luego cuando la instalación lo requiera creemos las nuevas particiones con "fdisk", al menos una partición Nativa Linux y una partición de intercambio para Linux, la partición de intercambio se utilizará para suministrar memoria virtual al sistema, y debe tener un tamaño entre 16 y 128 Mégabytes, dependiendo de cuanto espacio de disco tenga usted y cuantos programas quiera ejecutar. Linux no usa mas de 128 Megabytes de intercambio, asi que no hay razon para hacer su partición de intercambio mayor.

Se recomienda el uso de una partición de intercambio sea cual fuere su RAM, y si su sistema tiene mas de 16 mégabytes de RAM lo recomendable para un uso promedio del sistema es usar la mitad de RAM real como memoria de intercambio. La particion "Linux" contendrá todos sus archivos, y puede hacerla de cualquier tamaño entre 150 Mégabytes y el tamaño máximo del disco, menos el tamaño de la partición de intercambio.

Si usted esta familiarizado con Unix o Linux, puede querer crear particiones adicionales, por ejemplo, puede crear particiones que contendran los sistemas de archivos /var, y /usr.

Si ya tenemos listo el disco rígido y la información necesaria podemos proceder a la instalación para ello necesitaremos bootear desde MSDOS, para ello podemos utilizar un disco de inicio de

Windows e iniciar en modo compatibilidad con CD-ROM.

- Ingresar al directorio DOSUTILS
- Ejecutar AUTOBOOT.BAT el que se encargará de llamar a "LOADLIN.EXE" que es un programa DOS que carga una imagen del Kernel en RAM borrando todo vestigio del DOS. si posee una version de Linux DEBIAN el comando a ejecutar es BOOT.BAT, y cumple la misma función que el comando AUTOBOOT.BAT en su sistema.
- Luego del Booteo, Linux entra automáticamente en una aplicación que no es otra cosa que un asistente para la instalación. A él deberemos contestar las preguntas que nos vaya haciendo de acuerdo a la información que recopilamos.
- Tambien el asistente se encargará de preguntar acerca del nuestro disco rígido dándonos la posibilidad de ejecutar "fdisk" a fin de particionar nuestro disco, el uso de fdisk es sencillo a pesar de ser mucho más poderoso que el de DOS y Windows basta decir que con "m" accedemos a la ayuda del mismo, lo que hora debemos hacer es, crear la partición SWAP con por lo menos la mitad de la RAM real, para ello usaremos la opción "n" y luego de decirle el número de partición le indicaremos el tipo que deber ser primaria, con "t" le cambiamos el tipo a "82" que es SWAP. A continuación crear la partición Linux Nativa con "n" nuevamente que también deber ser primaria, el tipo no es necesario cambiarlo, por cuanto por defecto crea particiones Linux Nativa (tipo 83).
- Seleccionar los paquetes que instalaremos, esta es una lista muy extensa, son imprescindibles los paquetes base, C, C++, X11, Network, pero son necesarios casi todos si lo que pretendemos es un servidor de RED, lo superfluo son lo juegos y algunas aplicaciones muy especificas como el diseño de circuitos y microprocesadores. Es recomendable instalar el Midnight Commander que es un clon de Norton Comander para Linux el que se encuentra en el paquete de Shells y el "joe" que es un editor de texto.
- Ahora el asistente nos ayudara a instalar "lilo" que es el sistema de arranque dual de Linux el que nos permitirá la posibilidad de arrancar con mas de un Sistema Operativo.

☐ La estructura o sistema de archivos de linux se denomina Ext2, no es compatible con tipos de estructuras FAT, FAT32 o NTFS, lo que signufica que no es accesible desde sistemas operativos Windows como una unidad fisica.

[Indice General](#)

Comandos de Linux

Es importante destacar que todos los nombres de archivos y comandos son "case-sensitive" (que hacen diferencia entre mayúsculas y minúsculas, a diferencia de sistemas operativos como MS-DOS). Por ejemplo, el comando `make` es diferente a `Make` o `MAKE`. Lo mismo ocurre en el caso de nombres de archivos o directorios.

Cambio de directorio

La orden para movernos por la estructura de directorios es `cd`, abreviacion de "change directory" o cambio de directorio. Hay que destacar, que la mayoría de las órdenes UNIX mas usadas son de dos o tres letras

La forma de uso de la orden `cd` es: **`cd "directorio"`**

donde "directorio" es el nombre del directorio al que queremos ir. Si usamos `cd` sin parámetros esto nos llevará a nuestro directorio. Tenemos tambien dos punteros uno al directorio padre `..` y otro a sí mismo `.` Esto tambien existe en el comando DOS

Consultar el directorio actual

Una vez que podemos navegar por nuestro arbol de directorio lo mas importante es poder ver que tenemos en cada uno. Para eso usaremos el comando `ls`, `ls` vendría a ser "dir" del DOS, pero como veremos mas adelante; es mucho mas poderoso como todos los comandos de Linux, `ls` tiene muchos parámetros que van a afectar su comportamiento a todos, pero el mas usado sin duda ser `-l` que nos mostrará con mas detalle el contenido del directorio.

Consultar las paginas del manual

En este momento creo que uno de los comandos que más nos van a ayudar en el aprendizaje de Linux es "man" que siguiendo con las comparaciones con el DOS vendria a ser el "help" con man podremos solicitar información de todos los comandos e incluso algunos programas y aplicaciones tiene sus propias páginas de manual. En mi experiencia personal man me ayudo mas veces que cualquier libro

La forma de usarlo es: **`man "comando"`**

Creacion de directorios

Para esto usaremos el comando `mkdir` como en DOS seguido del nombre del directorio a crear

Se aplica de la siguiente manera: **`mkdir "nombre"`**

Copiado de archivos

La orden "cp" copia los archivos listados en la linea de comandos al archivo o directorio pasado como último argumento. Nótese como se usa el directorio "." para referirnos al directorio actual

La forma de uso de la orden cp es: ***cp "origen" "destino"***

Mover archivos

Tanto para mover como para renombrar archivos o directorio usaremos el comando "mv" de idéntica manera a "cp" solo que en esta caso no se preservará el origen.

Se aplica de la siguiente manera: ***cp "origen" "destino"***

Borrado de archivos

Para borrar se usa el comando "rm" que viene de "remove". Se usa con un parámetro que es el nombre del archivo a borrar o una lista de ellos. Tambien lo podemos usar para borrar directorios completos con el parámetro "-r" hay que ser muy prudente y cuidadoso con este comando.

Forma de uso: ***rm "archivo o directorio"***

Mirando los archivos

Las órdenes "more", "cat" y "less" son usadas para ver el contenido de archivos. More y less muestran el fichero pantalla a pantalla mientras que cat lo muestra entero de una vez

- more "archivo"
- cat "archivo"
- less "archivo"

Montando unidades

A diferencia del Sistema Operativo DOS, o de Windows, Linux al ser un Sistema Operativo basado en un solo núcleo no posee unidades lógicas asignadas por letras "A:", "C:", "D:", para poder acceder a otras unidades que posea nuestra computadora como puede ser una unidad de disquettes o una lectora de CDROM, debemos montarla dentro de nuestro sistema. Esto parece algo muy complejo a simple vista pero no es así. Existe un comando llamado "mount" que cumple la función de montar unidades dentro del sistema.



Por lo general usted encontrará en cualquier versión de Linux un directorio llamado ***"/mnt"*** derivación de la palabra "mount" y dentro de este dos subdirectorios llamados ***"/mnt/floppy"*** donde se ubicarán los disquettes montados para su lectura y otro subdirectorio llamado ***"/mnt/cdrom"*** donde se montará la unidad de CDROM.

La forma de montar unidades es: ***mount "unidad" "directorio destino"***

Supongamos que usted quiere leer el contenido de un disquette, entonces debera hacer lo siguiente:

- Inserte el disquette en la unidad de disquettes.
- Desde la consola una terminal tipee: **"mount /dev/fd0 /mnt/floppy"**
- Presione "Enter" para confirmar el comando.

☐ **"/dev/fd0" equivale en DOS a la unidad A:, si posee una segunda unidad de disquettes entonces esta se llamara "/dev/fd1" o el equivalente en DOS a B:.**

Ahora bien si usted desea montar su unidad de CDROM, debera seguir los pasos anteriormente mencionados pero asignando esta vez **"/dev/cdrom"**.

Ejemplo: **mount /dev/cdrom /mnt/cdrom**

Desmontando unidades

Otra característica que se diferencia de Sistemas Operativos como el DOS o Windows es que si necesitamos leer el contenido de otro disquette por ejemplo y ya hemos montado uno anteriormente debemos desmontar el primero para poder leer el siguiente.

Esto se realiza con el comando "umount" que desmonta las unidades preparando al sistema para el montaje de una nueva unidad.

La forma de desmontar una unidade es: ***umount "unidad"***

Ejemplo, para desmontar una unidad de disquettes tipéo la siguiente órden:

umount /dev/fd0

[Indice General](#)

Estructura Del Sistema

Al ser un sistema basado en un núcleo GNU/Linux cuelga de una raíz (root system), su punto de partida es "/" por eso al instalarlo nos solicita que definamos el punto de montaje o sea, desde donde "colgará" el mismo.

Dentro de él hay una serie de directorios con diferentes permisos, algunos exclusivos para el administrador y otros de dominio público.

A continuación haré una breve descripción de los directorios principales del sistema.

drwxr-xr-x	2	root	root	4096	Jun 18 12:04	bin
drwxr-xr-x	3	root	root	1024	Jul 29 12:07	boot
drwxr-xr-x	11	root	root	98304	Jul 29 12:11	dev
drwxr-xr-x	42	root	root	4096	Jul 29 12:25	etc
drwxr-xr-x	5	root	root	4096	Jul 10 14:58	home
drwxr-xr-x	4	root	root	4096	Jun 3 11:49	lib
drwxr-xr-x	2	root	root	16384	Jun 3 11:46	lost+found
drwxr-xr-x	4	root	root	4096	Jul 26 16:32	mnt
drwxr-xr-x	2	root	root	4096	Aug 23 1999	opt
dr-xr-xr-x	115	root	root	0	Jul 29 08:07	proc
drwxr-x---	19	root	root	4096	Jul 29 12:28	root
drwxr-xr-x	2	root	root	4096	Jun 10 15:10	sbin
drwxrwxrwt	9	root	root	4096	Jul 29 12:29	tmp
drwxr-xr-x	17	root	root	4096	Jul 16 15:11	usr
drwxr-xr-x	23	root	root	4096	Jun 3 17:27	var

Fig003, Listado del arbol del sistema desde una terminal en modo texto.

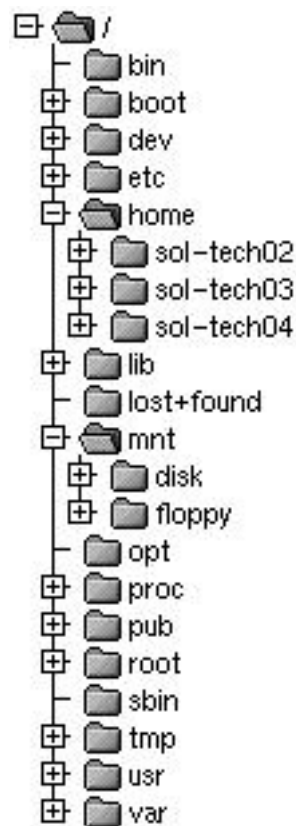


Fig004, Listado del árbol del sistema desde modo gráfico (Xwindow)

/bin

Directorio que contiene archivos ejecutables del sistema de formato binario.

/boot

Contiene toda la información necesaria para el inicio del sistema.

/dev

Es el directorio donde se encuentran todos los dispositivos del sistema, específicamente se refiere al hardware. Aquí se encuentran listados todos los dispositivos soportados por el sistema por default "*definición de programa*", unidades de almacenamiento, impresoras, escaners, discos ópticos, unidades tape backup, modems, etc..

/etc

Toda configuración queda almacenada dentro de este directorio. Servidores http, irc, de conectividad "[Samba](#)", shells del sistema, etc. Además se encuentran subdirectorios que contienen los scripts de inicio del sistema los cuales nos permiten ejecutar demonios cuando el sistema inicia. Se denominan demonios "**daemons**" a aquellos programas que se ejecutan en segundo plano para realizar alguna función específica. Podemos mencionar al demonio del servidor de intercambio de archivos Samba llamado *smbd* el cual se ejecuta al inicio del sistema y nos permite conectarnos con clientes de otros sistemas operativos y que puedan acceder al servidor.

/home

Su traducción explícita al castellano sería **hogar** se refiere a los *hogares* de los clientes de nuestra red. Por ejemplo supongamos que en nuestro sistema tenemos asignado un usuario de nombre "sol-tech02" entonces su hogar será `/home/sol-tech02` Este usuario al logearse tiene como punto de montaje el directorio anteriormente mencionado.

/lib

Contiene todas las librerías del sistema necesaria para la ejecución de programas, en su mayoría los programas dependientes de librerías son aquellos que se ejecutan bajo entorno gráfico. Este directorio sería el similar a el directorio "system" de Microsoft Windows, donde se encuentran las librerías .dll necesarias para la ejecución de programas.

/lost+found

Si por alguna razón se produce un error en el sistema, un programa comete un error crítico o se pierde referencia de información dentro del sistema. Se volcará un archivo log dentro del mismo detallando el error.

/mnt

Al ser un sistema basado en un solo núcleo, Linux necesita tener definido un directorio donde montar unidades. En la versión de Linux RedHat se encuentran por "*default*" definidos dos subdirectorios "`/mnt/cdrom`" y "`/mnt/floppy`".

/opt

Todo proceso tiene una "*PID*" identificación de proceso. Aquí se generan todos los eventos relacionados con los PID. Y los procesos de los demonios o de programas ejecutados en segundo plano se almacenan en el directorio "*/proc*".

/root

El hogar del administrador del sistema. Dentro se almacenarán logs, configuraciones personalizadas, etc. Este directorio es restringido para los demás usuarios, eso significa que solo el administrador puede ingresar a este directorio.

/sbin

Directorio que contiene archivos de ejecución en formato binario.

/tmp

Aquí se generan archivos temporarios que son descartados por el sistema al finalizar la sesión.

/usr

Usr es una abreviación de *user* en inglés usuario. Contiene subdirectorios varios con documentación, programas, archivos ejecutables, etc.. Es un directorio de acceso general,

significa que cualquier usuario habilitado al sistema puede ingresar sin restricciones.

/var

Contiene directorios que a menudo cambian su tamaño o tienden a crecer. Muchos de estos directorios solían residir en /usr, por ejemplo en "/var/adm" encontraremos varios archivos de interés para el administrador del sistema, específicamente históricos del sistema, los cuales recogen errores o problemas con el sistema.

Manejo De Sesiones

Existen diferentes métodos para poder conectar las terminales al sistema:

- En primer lugar podemos conectarnos a un Linux a través de un puerto serie (RS232), con una terminal boba o bien con otro equipo y un emulador de terminales. En ambos casos existe un programa que atiende las solicitudes de conexión a través del puerto serie. Cuando hay una solicitud de conexión, éste programa la atiende solicitando al usuario que se identifique ante el sistema. Cuando termina la conexión, éste programa se reactiva para seguir atendiendo nuevas solicitudes.
- Con una placa de red. En este caso, tenemos un programa que escucha las solicitudes de conexión a través de la placa de red. Cuando llega una solicitud este programa se desdobra de forma que una parte atiende la conexión y otra continúa atendiendo nuevas conexiones. Así podemos tener más de una conexión a través de la placa de red. Algunos servicios que proveen esta funcionalidad son el sistema telnet (sin encriptación de datos) y el ssh (Secure Shell, con encriptación de datos).
- La consola, en un sistema GNU/Linux también puede ser activada para trabajar desde el teclado y monitor al que estén conectados directamente al sistema. Normalmente en la mayoría de las distribuciones, en la consola hay hasta 6 terminales virtuales, accediendo a cada una de ellas con las teclas

Alt+F1 a **Alt+F6**

Una vez que se ha conseguido conectar a un sistema Linux tenemos que iniciar una sesión de trabajo. Linux es un sistema multiusuario, y esto exige que el usuario se presente al sistema y que este lo acepte como usuario reconocido. Así cada vez que iniciamos una sesión Linux nos responde con:

Login:

A lo que se debe responder con el nombre de usuario. Luego, Linux solicita una clave para poder comprobar que el usuario es quien dice ser:

Password:

En este caso se teclea la clave de acceso. Por motivos de seguridad esta clave no aparecera en la pantalla. Si la pareja nombre de usuario/clave es correcta el sistema inicia una terminal en modo texto con la que se puede trabajar.

Habitualmente sera el simbolo "\$", aunque puede ser tambien el simbolo "%" (si usamos un shell C). Cuando es el administrador (root) quien esta trabajando en el sistema, el indicador que aparece es "#".



Consola de inicio en este caso version Linux Mandrake

El archivo passwd

Para iniciar una sesion de trabajo en un sistema Linux se debe suministrar al sistema una pareja de nombre de usuario/clave. Estos datos se almacenan en un archivo llamado **passwd**. Este archivo se hubica en el directorio /etc y contiene una linea por cada usuario del sistema. Cada linea consta de una serie de campos separados por dos puntos (:). Estos campos son, en el orden que aparecen:

- Nombre de usuario. Es es nombre con el que nos identificamos al sistema, con el que tenemos que responder al Login: y por el que nos identifica el sistema.
- Clave cifrada. El siguiente campo es la clave de acceso al sistema. Esta clave no se guarda como se introduce, sino que se almacena transformada mediante el algoritmo DES para que nadie pueda averiguarla.
- UID. *Identificador de usuario*. Es el número de usuario que tiene cada cuenta abierta en el sistema.
- GID. *Identificador de grupo*. Es el número de grupo principal al que pertenece el usuario.
- Nombre completo de usuario. Este es un campo informativo, en el que se suele poner el nombre completo del usuario.
- Directorio personal. Este campo indica el directorio personal de un usuario, en el cual el usuario puede guardar su información.
- Interprete de comandos. El último campo indica un programa que se ejecutará cuando el usuario inicie una sesión de trabajo. Normalmente este campo es una shell que proporciona una linea de ordenes para que el usuario trabaje. Ejemplo:

```
sol-tech03:x%9rTMn$Us:502:502:Facundo Gimenez:/home/sol-tech03:/bin/bash
```

Numero de usuario clave cifrada UID GID Nombre de usuario (completo) Directorio Shell

```
root:$1$7EEJ4.qu$Y3XMPsW1Wv5Tyr5PuG9Xw1:11849:0:99999:7:::
bin:*:11841:0:99999:7:::
daemon:*:11841:0:99999:7:::
adm:*:11841:0:99999:7:::
lp:*:11841:0:99999:7:::
sync:*:11841:0:99999:7:::
shutdown:*:11841:0:99999:7:::
halt:*:11841:0:99999:7:::
mail:*:11841:0:99999:7:::
uucp:*:11841:0:99999:7:::
ftp:*:11841:0:99999:7:::
xfs:!!:11841:0:99999:7:::
mailnull:!!:11841:0:99999:7:::
apache:!!:11841:0:99999:7:::
named:!!:11841:0:99999:7:::
sol-tech03:$1$qAKm0sfJ$YsmMIbTFfv79Dht8AQgsz1:11841:0:99999:7:::
sol-tech02:$1$atGzT0sJ$ejDLvUZ3JxsexF9MZD8x6.:11841:0:99999:7:::
```

Fig00c: El archivo passwd cifrado se vuelca al archivo shadow

[Indice General](#)

Sistema de permisos

Como he mencionado anteriormente el sistema Linux es un sistema multitarea/multiusuario. A diferencia del sistema operativo MSDOS (por ejemplo), este sistema operativo tienen la capacidad de tener a más de un usuario dentro del sistema, tanto localmente como remotamente. Por lo tanto para proteger la privacidad de los archivos y directorios de cada uno de ellos, para proteger ciertos archivos elementales para el buen funcionamiento de la máquina, y para una buena organización del sistema (aparte de otras muchas razones), tenemos a nuestra disposición los llamados permisos.

Como ya sabrá, en una computadora con el Sistema Operativo MSDOS, cualquier persona que acceda a ella puede en cualquier momento borrar todo el disco duro. En los sistemas basados en UNIX hay diferentes categorías de usuarios, siendo la más alta la del llamado "root", que puede leer, escribir y ejecutar en cualquier archivo dentro del sistema. Como he citado antes, hay 3 posibilidades a la hora de "tratar" a un archivo. Leer el archivo, ejecutar el archivo y escribir en el archivo (por supuesto, nada de esto podremos hacer si no tenemos permiso para ello o no hemos establecido los permisos oportunos).

Toda persona que haya tenido la oportunidad de trabajar con un sistema basado en UNIX, si ha tenido la ocasión de observar los archivos de passwords, habrá visto que aparte de la información del usuario, login, definición de shell se encuentran dos números, para ser más exactos, en el tercer y cuarto campo. El tercero UID, y el cuarto, también definido en /etc/group GID. Todo esto a groso modos, serviría para identificar a cada archivo con su propietario y grupo, entre otras cosas.

Los permisos de los archivos se almacenan junto con otra información en una palabra de 16 bits dentro de lo llamado "i-node" nueve de ellos, se refieren a lo mencionado anteriormente (lectura, escritura y ejecución), además de 3 bits adicionales que contienen cierta información para las operaciones con el archivo si es ejecutable, que ya explicaremos más adelante. Cada archivo posee un "i-node" diferente, que además de contener todo lo que hemos citado anteriormente, este número también hace referencia a los X primeros bloques en el disco donde se encuentra ese archivo, y vienen las referencias de los bloques que contienen las direcciones simples, dobles y triples del resto de los bloques del archivo.

En los archivos, de Linux, a la hora de establecer los permisos deseados tenemos que hacer uso de una orden de este S.O, que ha sido destinada para eso exactamente, para la especificación de permisos, esta orden se llama "chmod". A la hora de establecer los permisos, tenemos dos posibilidades, las dos son igual de válidas y buenas, aunque cada uno siempre se habituará a la que le resulte más cómoda. Las dos posibilidades citadas anteriormente reciben el nombre de absoluta y relativa, que más adelante hablaremos sobre ellas. A continuación explicaremos las características globales que tienen los permisos de los archivos tanto hayan sido asignados en forma absoluta como relativa.

Fijese en la siguiente figura:

Fig 1.

```
-rwxrwxrwx 1 core users 1929 May 15 23:00 prueba*
```

Gracias al comando `ls -l` podemos obtener bastante información tanto de los archivos como de los directorios del sistema Linux, tras hacer un `ls -l` prueba el sistema nos mostrará algo parecido a la Figura 1. Que como podemos observar el usuario "core" del grupo "users" tiene un archivo llamado "prueba". Si observamos el primer fragmento de la línea, nos extrañará el ver repetidamente las letras r,w y x, además del - al principio de todo. Pues bien como hemos citado antes, un S.O basado en UNIX tiene diferentes categorías de usuarios, es por ello que en el archivo "prueba" y en todos los archivos, se engloban todos estos grupos.

El primer "-" significa que es un archivo normal, ya que si fuese un directorio dicho "-" nos lo representaría por "d", si fuese un enlace mediante "l", "b" significaría que es un dispositivo especial de bloques, "c" un dispositivo especial de caracteres, "p" un archivo de conducción por nombre, "s" un semáforo y "m" nos indicaría que es un archivo de memoria compartida.

Las 3 primeras letras 'rwx' si nos fijamos en la Fig 1. sabemos que se refiere tanto a permisos de lectura, escritura y ejecución, pero este primer terceto siempre se referirá al usuario que creó dicho archivo, es decir al usuario "core", por lo tanto sabemos que este usuario puede leer, escribir y ejecutar dicho archivo. Pero, ¿Que hay del resto? Pues bien, el segundo terceto siempre se referirá a los permisos del grupo, mientras que lo que resta a los demás usuarios. Por lo tanto en este caso, todo el mundo tiene derecho a todos los permisos posibles sobre el archivo. Pongamos un último ejemplo:

```
-r-x--xrw- 1 core users 1929 May 15 23:00 prueba*
```

En este archivo como se puede observar hay un ligero cambio de permisos. El archivo "prueba" ya que comienza mediante un "-" tiene permisos de lectura y ejecución para su dueño "core", de ejecución para los de su grupo "users", y de lectura y escritura para el resto de usuarios. Como habrá podido ver, al no haber permisos de escritura para su dueño, ni permisos de lectura y escritura para su grupo, ni permisos de ejecución para el resto de los usuarios, estos son representados mediante un "-".

Ahora que ya he explicado las características globales de todo directorio y archivo unix daremos paso a la explicación de las dos formas citadas anteriormente para establecer permisos en Linux

Forma absoluta:

La principal característica que tiene esta nomenclatura es la de asignar los permisos mediante valores octales. La base octal, o base 8, puede contener de los números 0 al 7. Es por ello, que tan solo son válidos estos números a la hora de asignar permisos. Fíjese en el siguiente cuadro:

Fig 1. Permiso Valor_Octal

```
-----
r | lectura 4
w | escritura 2
```


x | ejecucion 1

Claramente se observa que número octal corresponde a cada característica de un archivo o directorio en un sistema UNIX. Como mencione antes el comando utilizado por los Sistemas UNIX para establecer permisos recibe el nombre de `chmod`. A la hora de dar permisos en forma absoluta hemos de seguir la siguiente Sintaxis: "`chmod XYZ archivo`".

X representa al dueño del archivo, Y al grupo y Z al resto de usuarios, mientras que "archivo" es el nombre del archivo a especificar los permisos. Siempre que queramos atribuir mas de un permiso los numeros octales se sumarán. Observar los siguientes ejemplos para un mayor entendimiento:

```
chmod 460 archivo
chmod 755 archivo
chmod 050 archivo
chmod 000 archivo
chmod 777 archivo
```

EL primer ejemplo daría permisos de lectura al propietario (4), lectura y escritura al grupo (4+2), y ningún permiso al resto de usuarios (0).

El segundo ejemplo daría permisos de escritura, lectura y ejecución al propietario (4+2+1), permisos de lectura y ejecución tanto al grupo como al resto de usuarios (4+1)

El tercer ejemplo no daría ningún permiso al propietario (0), permisos de ejecución y lectura al grupo (4+1), y ningún permiso al resto de usuarios (0)

El cuarto ejemplo no tendría ningún permiso para nadie (0). Y finalmente el último ejemplo tendría todos los permisos para todos los usuarios (4+2+1)

Forma relativa:

A diferencia de la forma absoluta, la relativa no utiliza números octales o en base 8 para establecer los diferentes permisos, sino que se basa en una nomenclatura de letras. Al igual que en la forma absoluta, en la relativa también nos valemos del comando "`chmod`" para asignar los permisos.

Observe los siguientes cuadros:

Fig 1.

Letra

- a**, Engloba todos los usuarios, grupos y demás usuarios.
- g**, Engloba el grupo del propietario.
- o**, Engloba todos los demás usuarios no mencionados antes.
- u**, Engloba al usuario que creo dicho archivo

Fig 2.

Operador

+ Agrega la modalidad

- Elimina la modalidad

= Elimina los permisos existentes y agrega los establecidos que sentencia que indiquemos.

Fig 3.

Permiso

x, Establecemos la ejecución

r, Establecemos la lectura

w, Establecemos la escritura

Siempre que se quiera atribuir permisos a un archivo/directorio en forma relativa se debe seguir el orden de los tres cuadros expuestos anteriormente. Por lo tanto primero se indicará a la persona o personas a la que se quiere atribuir dichos permisos, ejemplo:

chmod a+r archivo

Si se ha fijado en las Figuras anteriores, no te debería costar demasiado entender los permisos que atribuye la orden anterior. Primeramente como he dicho, se debe indicar al usuario/usuarios, en este caso esta la letra "a" que significa que se establecen permisos al dueño, grupo y demas usuarios. El siguiente simbolo que le sigue, indicará si se quiere agregar o eliminar permisos del archivo en cuestión, en este caso el simbolo "+" indica que deseo agregar permisos al archivo. Finalmente se indicarán los permisos a agregar. En este caso se indica "r" (lectura) para todos los usuarios "a".

Si por el contrario se quiere establecer varios permisos a las diferentes categorias que comprenden los sistemas UNIX se pueden utilizar las comas.

Ejemplo:

chmod o-wr, g-wr archivo

Que simplemente esta sentencia eliminaría ambos permisos, de escritura y lectura tanto para el grupo, como para el resto de usuarios o general.

[Indice General](#)

El Kernel

El kernel de UNIX actúa como intermediario entre los programas, la computadora y el usuario. En primer lugar, gestiona la memoria de todos los programas o procesos, y asegura que se asignen ciclos del microprocesador a cada uno. Además, proporciona una interfaz portable para que los programas hablen fácilmente con su hardware.

¿Para que recompilar el kernel?

Las nuevas versiones normalmente ofrecen la posibilidad de entenderse con más dispositivos de hardware (o sea, incluyen más drivers), se ejecutan más rápidamente, son más estables o corrigen errores de otras versiones. Además la recompilación nos permite adecuar el núcleo a el hardware que tenemos pudiendo de esta manera acceder por ejemplo a placas de sonido, video, etc. Otra de las ventajas es que podemos seleccionar que sistemas de archivos vamos a necesitar, por ejemplo en mi caso únicamente accedo a sistemas Vfat de Windows, MS-DOS, ISO9660, y Ext2fs, de manera que no ocupo innecesariamente recursos al incluir sistemas de archivos por ejemplo para OS/2 o Coherent.

¿Que necesito para recompilar el Kernel?

Necesitaremos:

- Las fuentes del Kernel que se proveen con cualquier distribución de Linux, también los podemos actualizar de Internet para poder contar con la última versión.
Aproximadamente los fuentes comprimidos con gzip ocupan 8MB y una vez descomprimido en el disco usan cerca de 30MB.
- El compilador de C "gcc" incluido en el paquete de "Development"
- Las librerías "libc"

Entre en la cuenta root y vaya al directorio /usr/src. Si instaló las fuentes en la instalación de Linux, habrá aquí un directorio linux, con las fuentes antiguas. Si quiere preservar esas fuentes (y tiene espacio de sobra), renombre (con mv) ese directorio a linux-x.y.z. Es importante que antes de descomprimir las nuevas fuentes no haya ningún directorio /usr/src/linux.

Ahora, en el directorio /usr/src, descomprima las fuentes con:

```
tar xvfzp linux-x.y.z.tar.gz
```

Configuración del Kernel

Antes de recompilar debemos configurar que características tendrá nuestro nuevo kernel para ello primero limpiaremos de anteriores configuraciones el disco desde la terminal con el comando:

make mrproper

Luego comenzaremos con la configuración en sí desde la terminal con el comando:

make config

Invocado en el directorio /usr/src/linux inicia un programa shell script de configuración que le preguntará bastantes cosas. Requiere que esté instalado bash, con lo que debe existir /bin/bash, /bin/sh o la variable \$BASH.

Si las fuentes que posee son superiores al 2.0.x, existen dos métodos bastante mas agradables e interactivos de llevar a cabo la configuración, invocando desde la terminal de modo texto, en el mismo directorio mencionado anteriormente

make menuconfig

Inicia un colorido programa muy agradable y bastante mas funcional que el make config a secas, para llevar a cabo la configuración del kernel.

☐ Observar que su uso es bastante intuitivo; las opciones con un < > indican la posibilidad de modularizar; si en ella presionamos "M" la seleccionaremos como modulo. Si presionamos ESPACIO aparecera un "*" indicando que sera incluido en el kernel. Finalmente, si presionamos "N" lo deseleccionaremos.

Si usted desea ejecutar el menu en modo gráfico, entonces puede arrancar desde una terminal el script equivalente, ejecutando

make xconfig

A continuacion describire las opciones que creo son mas importantes a tener en cuenta al momento de compilar de todos modos las que aquí no esten comentadas y usted desée conocer cualquiera de los metodos de configuración usados tendrá un acceso a ayuda.

Emulación de coprocesador (*kernel math emulation*): Si no tiene coprocesador matemático debe contestar "y" a ésto. Si tiene coprocesador y contesta "y" la emulación será instalada pero no usada; funcionará, pero le ocupara memoria innecesariamente. Además, he oído que la emulación es lenta.

Soporte de discos IDE y MFM/RLL: Este soporte es necesario normalmente, pues es el soporte de los discos habituales en las computadoras. Este manejador no gestiona discos SCSI, para eso esta la opción específica que luego veremos.

Soporte de redes (*networking support*): En principio, contestaría afirmativamente si la máquina estuviera en una red local o conectada a Internet con conexión SLIP/PPP. Sin embargo, muchos programas, como ocurre con el sistema X Window, requieren el soporte de

red, así que lo mejor es contestar "y" siempre. Después le preguntará si quiere soporte TCP/IP: conteste afirmativamente.

Limitar memoria a 16 Mb (*Limit memory to low 16MB*): Hay controladores de DMA en equipos 386 con problemas para direccionar más de 16 Mb de RAM. Conteste "y" en caso (raro) de que tenga uno de estos motherboards.

Comunicación entre procesos System V (*System V IPC*): Sinceramente no sé bien lo que es pero leí que no es recomendable desactivarlo.

Tipo de procesador (*Processor type*): Es recomendable elegir siempre el CPU que tenga, aunque la versión de 386 suele utilizarse para compatibilizar el medio de transferencia en equipos con problemas al momento de compilar el kernel.

Soporte SCSI (*SCSI support*): Si tiene dispositivos SCSI, conteste "y". Se le preguntará entonces, opciones más específicas sobre el dispositivo soporte, si él es un CD-ROM, disco rígido, unidad de almacenamiento óptica, etc. Y que clase de adaptador utiliza.

Soportes de placas de red (*Network device support*): Si tiene una placa de red, o quiere usar una conexión PPP, SLIP o PLIP, conteste afirmativamente. Luego de afirmar la pregunta se le preguntará que placa de red tiene y cuáles son los protocolos a usar.

Sistemas de archivos (*Filesystems*): El programa le preguntará si quiere soporte para los siguientes sistemas de archivos:

- Standard (minix), Las nuevas distribuciones no usan sistemas de archivos Minix, y la mayoría de la gente ya no utiliza este tipo de sistema. No necesita seleccionarla.
- Extended fs, Fue la primera versión del sistema extendido, pero ya no se usa. No necesita seleccionarla.
- Second extended, Este es el sistema más usado, conocido como Ext2 es el sistema de archivos nativo de Linux, así como el sistema FAT lo es en DOS. Seleccione "y"
- xiafs filesystem, Sistema de archivos antiguo, No necesita seleccionarlo
- msdos, Selecciónelo si quiere acceder desde Linux a sus particiones de MS-DOS o quiere montar disquetes de ese sistema.
- umsdos, Este sistema permite añadir a un sistema MS-DOS las características típicas de Unix como nombres largos o enlaces. Sirve para quienes quieran instalar Linux en la partición DOS, pero nada más.
- /proc, (una idea robada descaradamente a Bell Labs, si no me equivoco). No se trata de algo que se guarde en el disco, sino de una interfaz por medio de archivos con la tabla de procesos del núcleo, usada por comandos como "ps". Pruebe a teclear "cat /proc/meminfo" o "cat /proc/devices". Algunos shells (el shell rc, concretamente) hacen uso de /proc/self/fd (en lugar de /dev/fd) para el manejo de Entrada y Salida. Debe contestar afirmativamente a la pregunta: muchas utilidades de Linux lo necesitan
- NFS, Si su máquina está en red y quiere acceder a sistemas de archivos remotos con NFS, conteste "y" a esta pregunta.

- ISO9660, Sistema de archivos típico en los CD-ROMs. Si tiene un CD-ROM, seleccione esta opción.
- OS/2 HPFS, Para acceder a disco de OS/2. Es un sistema antiguo, no seleccionar
- System V and Coherent, para particiones de sistemas Coherent y System V (otras variantes de Unix para PC). Conteste "no".
- quota support, Selecciónelo si quiere administrar el espacio que consumen en disco los usuarios del sistema
- vfat, En conjunción con el sistema de archivos FAT o DOS nos permitirá acceso a las particiones de Windows 95, con soporte de nombres "largos".
- SMB filesystem support, En conjunción con el paquete samba nos permitirá montar volúmenes compartidos por clientes de red NetBios
- Amiga FFS support, Soporte para el sistema de archivos Fast File System de Amiga
- UFS filesystem support, Soporte para el sistema de archivos de UNIX, empleado por algunos Unix BSD, como SunOS, FreeBSD, NetBSD, NeXTstep. Permite montar dichas particiones o disquetes en modo de sólo lectura.
- NCP filesystem support, Permite montar volúmenes NetWare. Necesitará algún programa cliente para llevarlo a cabo

Manejadores de tipo caracter (*Character devices*): Aquí se activan los drivers para impresoras (por puerto paralelo), mouses de bus, mouses PS/2 (usado también en muchos trackballs de notebooks), algunos drivers de cinta y etc. Conteste según su hardware

Placa de sonido (*Sound card*): Si tiene una placa de sonido conteste "y".

Otras opciones de configuración: No todas las opciones se han listado ya que cambian mucho y otras son completamente evidentes (por ejemplo, el soporte 3Com 3C509)

Kernel hacking: La configuración "kernel hacking" suele dar lugar a un núcleo más grande y/o lento, y puede hacerlo menos estable al configurar algunas rutinas especiales para encontrar errores del núcleo. Deber contestar "n" a la pregunta. En caso de afirmar la pregunta lea detalladamente la documentación de la opción.

El archivo "Makefile"

Tras hacer "make config", un mensaje le dirá que ya está preparado el kernel y que revise el archivo Makefile para opciones adicionales. Por lo tanto, veamos el Makefile. Probablemente no tendrá que cambiar nada, pero debe mirarse. Sus opciones podrán cambiarse con el comando "rdev" una vez que el kernel este compilado.

Limpieza y dependencias

Cuando se termina de configurar debe ejecutar "make dep" y "make clean". Haciendo "make dep", se prepararán las dependencias en poco tiempo, a menos que su PC sea muy lenta. Cuando acabe este proceso, tipee "make clean". Esto elimina archivos obsoletos. No olvidar este paso

La compilacion

Despues de preparar las dependencias, puede ejecutar "make zImage"(esta es la parte que tarda mas tiempo), para compilar el nucleo y lo dejará comprimido en el subdirectorio arch/i386/boot/zImage junto a otros archivos. La "z" inicial indica que lo que obtenemos es una imagen comprimida del kernel, se descomprime automáticamente al ser ejecutado

Compilación de Módulos

Las órdenes "make modules" y "make modules_install" compilarán los módulos que hayamos seleccionado, y los instalarán. No olvidar ejecutar depmod -a en cuanto hayamos arrancado con dicho nucleo

Instalación del Kernel nuevo

La instalación se hará en el directorio "/boot" si lo que instalamos es RedHat, en otro caso se la hace en "/". Veamos ahora como hacer esto:

```
cd arch/i386/boot
chmod 400 zImage
rdev -R zImage 1
cp zImage /boot/nuevokernel
```

Sólo nos resta retocar el archivo lilo.conf que se encuentra en el directorio /etc

[Indice General](#)

Tareas y procesos

Control de Tareas es una utilidad incluída en muchos shells (incluidas Bash y Tcsh), que permite el control de multitud de comandos o tareas al momento. Antes de seguir, deberemos hablar un poco sobre los procesos.

Cada vez que usted ejecuta un programa, usted lanza lo que se conoce como proceso, que es simplemente el nombre que se le dá a un programa cuando se esta ejecutando. El comando "ps" visualiza la lista de procesos que se estan ejecutando actualmente, por ejemplo:

```
/home/sol-tech02# ps
```

PID	TT	STAT	TIME	COMMAND
24	3	S	0:03	(bash)
161	3	R	0:00	ps

La columna PID representa el identificador de proceso. La última columna COMMAND, es el nombre del proceso que se esta ejecutando. Ahora solo estamos viendo los procesos que esta ejecutando sol-tech02. Vemos que hay dos procesos, bash (Que es el shell que usa sol-tech02), y el propio comando ps. Como puede observar, bash se ejecuta concurrentemente con el comando ps. Bash ejecuto ps cuando sol-tech02 tecleo el comando. Cuando ps termina de ejecutarse (después de mostrar la tabla de procesos), el control retorna al proceso bash, que muestra el prompt, indicando que esta listo para recibir otro comando. Un proceso que esta corriendo se denomina tarea para el shell. Los términos proceso y tarea, son sinónimos. Sin embargo, se suele denominar "tarea" a un proceso, cuando es usado en conjunción con control de tareas, que es un rasgo del shell que permite cambiar entre distintas tareas.

Hay muchos mas procesos aparte de estos corriendo en el sistema, para verlos todos, teclearemos el comando "ps -aux". En muchos casos, los usuarios solo ejecutan un trabajo cada vez, que es el último comando que ellos teclearon desde el shell. Sin embargo, usando el control de tareas, usted podra ejecutar diferentes tareas al mismo tiempo, cambiando entre cada uno de ellos conforme lo necesite. Supongamos que esta usando un procesador de textos, y de repente necesita parar y realizar otra tarea, con el control de tareas, usted podria suspender temporalmente el editor, y volver al shell para realizar cualquier otra tarea, y luego regresar al editor como si no lo hubiese dejado nunca. Lo siguiente solo es un ejemplo, hay montones de usos prácticos del control de tareas.

Primer plano y Segundo plano

Un proceso puede estar en Primer plano o en Segundo plano "background". Solo puede haber un proceso en primer plano por vez, el proceso que esta en primer plano, es el que interactúa con usted, recibe entradas de teclado, y envia las salidas al monitor. (Salvo, por supuesto, que haya redirigido la entrada o la salida. El proceso en segundo plano, no recibe ninguna señal

desde el teclado por lo general, se ejecutan en silencio sin necesidad de interacción.

Algunos programas necesitan mucho tiempo para terminar, y no hacen nada interesante mientras tanto. Compilar programas es una de estas tareas, así como comprimir un fichero grande. No tiene sentido que se sienta y se aburra mientras estos procesos terminan. En estos casos es mejor lanzarlos en segundo plano, para dejar la computadora en condiciones de ejecutar otro programa.

Los procesos pueden ser suspendidos. Un proceso suspendido es aquel que no se está ejecutando actualmente, sino que está temporalmente parado. Después de suspender una tarea, puede indicar a la misma que continúe, en primer plano o en segundo, según necesite. Retomar una tarea suspendida no cambia en nada el estado de la misma; la tarea continuará ejecutándose justo donde se dejó.

Tenga en cuenta que suspender un trabajo no es lo mismo que interrumpirlo. Cuando usted interrumpe un proceso (generalmente con la pulsación de "ctrl-C"), el proceso muere, y deja de estar en memoria y utilizar recursos de la computadora. Una vez eliminado, el proceso no puede continuar ejecutándose, y deberá ser lanzado otra vez para volver a realizar sus tareas. También se puede dar el caso de que algunos programas capturan la interrupción, de modo que pulsando "ctrl-C" no se para inmediatamente. Esto se hace para permitir al programa realizar operaciones necesarias de limpieza antes de terminar. De hecho, algunos programas simplemente no se dejan matar por ninguna interrupción.

Envío a segundo plano y eliminación de procesos

Empecemos con un ejemplo sencillo. El comando "yes" es un comando aparentemente inútil que envía una serie interminable de "y-es" a la salida estándar. (En realidad es muy útil. Si se utiliza una tubería (o "pipe") para unir la salida de "y-es" con otro comando que haga preguntas del tipo sí/no, la serie de y-es confirmaría todas las preguntas.

```
/home/sol-tech02#
```

```
yes
```

```
y
```

```
y
```

```
y
```

```
y
```

```
y "ctrl-c"
```

La serie de "y-es" continuará hasta el infinito, a no ser que usted la elimine, pulsando la tecla de interrupción. También puede deshacerse de esta serie de y-es redirigiendo la salida estándar de yes hacia /dev/null, que es como una especie de "agujero negro" o papelera para los datos. Todo lo que usted envíe allí, desaparece: "/home/sol-tech02# yes > /dev/null". Ahora va mucho mejor, la terminal no se ensucia, pero el prompt de la shell no retorna. Esto es porque yes sigue ejecutándose y enviando esos inútiles "y-es" a "/dev/null". Para recuperarlo, pulse la tecla de interrupción.

Supongamos ahora que queremos dejar que el comando "yes" siga ejecutandose, y volver al mismo tiempo a la shell para trabajar en otras cosas. Para ello enviaremos a yes a segundo plano, lo que nos permitira ejecutarlo, pero sin necesidad de interacción.

Una forma de mandar procesos a segundo plano es añadiendo un caracter "&" al final de cada comando.

```
/home/sol-tech02# yes >/dev/null &
[1] 164
```

Como podrá ver, ha regresado a la shell. ¿Pero que es eso de "[1] 164"? ¿Se esta ejecutando realmente el comando yes?. "[1]" representa el número de tarea del proceso yes. La shell asigna un número a cada tarea que se este ejecutando. Como yes es el único comando que se esta ejecutando, se le asigna el número de tarea 1. El número "164" es el número de identificación del proceso, o PID, que es el número que el sistema le asigna al proceso. Ambos números pueden usarse para referirse a la tarea como veremos después.

Ahora usted tiene el proceso yes corriendo en segundo plano, y enviando constantemente la señal y hacia el dispositivo /dev/null. Para chequear el estado del proceso, utilice el comando interno de la shell jobs:

```
/home/sol-tech02# jobs
[1]+  Running                  yes >/dev/null &
```

También puede usar el comando "ps", como mostramos antes, para comprobar el estado de la tarea.

Para eliminar una tarea, utilice el comando "kill". Este comando toma como argumento un número de tarea o un número de ID de un proceso. Esta era la tarea 1, así que usando el comando :

```
/home/sol-tech02# kill %1
```

Ahora que ya hemos matado la tarea, podemos usar el comando jobs de nuevo para comprobarlo:

```
/home/sol-tech02# job
[1]+  Terminated              yes >/dev/null
```

La tarea está , en efecto, muerta, y si usa el comando jobs de nuevo, no mostrará nada. También podrá matar la tarea usando el número de ID de proceso (PID), el cual se muestra conjuntamente con el ID de tarea cuando arranca la misma. En nuestro ejemplo el ID de proceso es 164, así que el comando:

```
/home/sol-tech02# kill 164
```

es equivalente a

```
/home/sol-tech02# kill %1
```

No es necesario usar el "%" cuando nos referimos a una tarea a través de su ID de proceso.

Detención o parada y relanzamiento de tareas

Hay otra manera de poner una tarea en segundo plano. Usted puede lanzarlo como un proceso normal (en primer plano), pararlo, y después relanzarlo en segundo plano.

Primero, lance el proceso yes en primer plano como lo haría normalmente:

```
/home/sol-tech02# yes >/dev/null
```

De nuevo, dado que "yes" corre en primer plano, no debe retornar el prompt de la shell.

Ahora, en vez de interrumpir la tarea con "ctrl-C", suspenderemos la tarea. El suspender una tarea no la mata; solamente la detiene temporalmente hasta que Ud. la retoma. Para hacer esto usted debe pulsar la tecla de suspender, que suele ser "ctrl-Z".

```
/home/sol-tech02# yes > /dev/null
```

```
"ctrl-Z"
```

```
[1]+  Stopped                yes >/dev/null
```

```
/home/sol-tech02#
```

Mientras el proceso está suspendido, simplemente no se está ejecutando. No gasta tiempo ni recursos del sistema en la tarea. Sin embargo, usted puede retomar el proceso de nuevo como si nada hubiera pasado. Continuará ejecutándose donde se dejó. Para relanzar la tarea en primer plano, use el comando fg (del inglés "foreground").

```
/home/sol-tech02# fg yes >/dev/null
```

La shell muestra el nombre del comando de nuevo, de forma que tenga conocimiento de que tarea es la que ha puesto en primer plano. Pare la tarea de nuevo, con "ctrl-Z". Esta vez utilice el comando bg para poner la tarea en segundo plano. Esto hará que el comando siga ejecutándose igual que si lo hubiese hecho desde el principio con "&".

```
/home/sol-tech02# bg
```

```
[1]+  yes >/dev/null &
```

```
/home/sol-tech02#
```

Y tenemos de nuevo el prompt. El comando jobs debería decirnos que yes se esta ejecutando, y podemos matar la tarea con kill tal y como lo hicimos antes.

¿Como podemos parar la tarea de nuevo? Si pulsa "ctrl-Z" no funcionará, ya que el proceso esta en segundo plano. La respuesta es poner el proceso en primer plano de nuevo, con el comando fg, y entonces pararlo. Como puede observar podra usar fg tanto con tareas detenidas, como con las que esten segundo plano.

Hay una gran diferencia entre una tarea que se encuentra en segundo plano, y una que se encuentra detenida. Una tarea detenida es una tarea que no se esta ejecutando, es decir, que no usa tiempo de CPU, y que no esta haciendo ningún trabajo (la tarea aun ocupa un lugar en memoria, aunque puede ser volcada a disco). Una tarea en segundo plano, se esta ejecutando, y usando memoria, a la vez que completando alguna accion mientras usted hace otro trabajo. Sin embargo, una tarea en segundo plano puede intentar mostrar texto en su terminal, lo que puede resultar molesto si esta haciendo hacer otra cosa. Por ejemplo, si usted uso el comando

```
/home/sol-tech02# yes &
```

Sin redirigir "stdout" a "/dev/null", una cadena de "y-es" se mostrará en su monitor, sin que pueda interrumpirlo (no puede hacer uso de "ctrl-C" para interrumpir tareas en segundo plano). Para poder parar esas interminables "y-es", tendría que usar el comando fg para pasar la tarea a primer plano, y entonces usar "ctrl-C" para matarla. Otra observación. Normalmente, los comandos "fg" y "bg" actuan sobre el último proceso parado (indicado por un "+" junto al número de tarea cuando usa el comando jobs). Si usted tiene varios procesos corriendo a la vez, podría mandar a primer o segundo plano una tarea especifica indicando el ID de tarea como argumento de fg o bg, como en "/home/sol-tech02# fg %2" (para la tarea de primer plano número 2), o "/home/sol-tech02# bg %3 "

(Para la tarea de segundo plano número 3). No se pueden usar los ID de proceso con fg o bg. Ademas de esto, si usa el número de tarea por si solo, como "/home/sol-tech02# %2".

Solo queda recordarle que el uso de control de tareas es una utilidad de la shell. Los comandos fg, bg y jobs son internos de la shell. Si por algun motivo usted utiliza una shell que no soporta control de tareas, no espere disponer de estos comandos. Y ademas, hay algunos aspectos del control de tareas que difieren entre Bash y Tcsh. De hecho, algunas shells no proporcionan ningún control de tareas sin embargo, la mayoria de las shells disponibles para Linux soportan control de tareas.

[Indice General](#)

Sistema de permisos

Como he mencionado anteriormente el sistema Linux es un sistema multitarea/multiusuario. A diferencia del sistema operativo MSDOS (por ejemplo), este sistema operativo tienen la capacidad de tener a más de un usuario dentro del sistema, tanto localmente como remotamente. Por lo tanto para proteger la privacidad de los archivos y directorios de cada uno de ellos, para proteger ciertos archivos elementales para el buen funcionamiento de la máquina, y para una buena organización del sistema (aparte de otras muchas razones), tenemos a nuestra disposición los llamados permisos.

Como ya sabrá, en una computadora con el Sistema Operativo MSDOS, cualquier persona que acceda a ella puede en cualquier momento borrar todo el disco duro. En los sistemas basados en UNIX hay diferentes categorías de usuarios, siendo la más alta la del llamado "root", que puede leer, escribir y ejecutar en cualquier archivo dentro del sistema. Como he citado antes, hay 3 posibilidades a la hora de "tratar" a un archivo. Leer el archivo, ejecutar el archivo y escribir en el archivo (por supuesto, nada de esto podremos hacer si no tenemos permiso para ello o no hemos establecido los permisos oportunos).

Toda persona que haya tenido la oportunidad de trabajar con un sistema basado en UNIX, si ha tenido la ocasión de observar los archivos de passwords, habrá visto que aparte de la información del usuario, login, definición de shell se encuentran dos números, para ser más exactos, en el tercer y cuarto campo. El tercero UID, y el cuarto, también definido en /etc/group GID. Todo esto a groso modos, serviría para identificar a cada archivo con su propietario y grupo, entre otras cosas.

Los permisos de los archivos se almacenan junto con otra información en una palabra de 16 bits dentro de lo llamado "i-node" nueve de ellos, se refieren a lo mencionado anteriormente (lectura, escritura y ejecución), además de 3 bits adicionales que contienen cierta información para las operaciones con el archivo si es ejecutable, que ya explicaremos más adelante. Cada archivo posee un "i-node" diferente, que además de contener todo lo que hemos citado anteriormente, este número también hace referencia a los X primeros bloques en el disco donde se encuentra ese archivo, y vienen las referencias de los bloques que contienen las direcciones simples, dobles y triples del resto de los bloques del archivo.

En los archivos, de Linux, a la hora de establecer los permisos deseados tenemos que hacer uso de una orden de este S.O, que ha sido destinada para eso exactamente, para la especificación de permisos, esta orden se llama "chmod". A la hora de establecer los permisos, tenemos dos posibilidades, las dos son igual de válidas y buenas, aunque cada uno siempre se habituará a la que le resulte más cómoda. Las dos posibilidades citadas anteriormente reciben el nombre de absoluta y relativa, que más adelante hablaremos sobre ellas. A continuación explicaremos las características globales que tienen los permisos de los archivos tanto hayan sido asignados en forma absoluta como relativa.

Fijese en la siguiente figura:

Fig 1.

```
-rwxrwxrwx 1 core users 1929 May 15 23:00 prueba*
```

Gracias al comando `ls -l` podemos obtener bastante información tanto de los archivos como de los directorios del sistema Linux, tras hacer un `ls -l` prueba el sistema nos mostrará algo parecido a la Figura 1. Que como podemos observar el usuario "core" del grupo "users" tiene un archivo llamado "prueba". Si observamos el primer fragmento de la línea, nos extrañará el ver repetidamente las letras r,w y x, además del - al principio de todo. Pues bien como hemos citado antes, un S.O basado en UNIX tiene diferentes categorías de usuarios, es por ello que en el archivo "prueba" y en todos los archivos, se engloban todos estos grupos.

El primer "-" significa que es un archivo normal, ya que si fuese un directorio dicho "-" nos lo representaría por "d", si fuese un enlace mediante "l", "b" significaría que es un dispositivo especial de bloques, "c" un dispositivo especial de caracteres, "p" un archivo de conducción por nombre, "s" un semáforo y "m" nos indicaría que es un archivo de memoria compartida.

Las 3 primeras letras 'rwx' si nos fijamos en la Fig 1. sabemos que se refiere tanto a permisos de lectura, escritura y ejecución, pero este primer terceto siempre se referirá al usuario que creó dicho archivo, es decir al usuario "core", por lo tanto sabemos que este usuario puede leer, escribir y ejecutar dicho archivo. Pero, ¿Que hay del resto? Pues bien, el segundo terceto siempre se referirá a los permisos del grupo, mientras que lo que resta a los demás usuarios. Por lo tanto en este caso, todo el mundo tiene derecho a todos los permisos posibles sobre el archivo. Pongamos un último ejemplo:

```
-r-x--xrw- 1 core users 1929 May 15 23:00 prueba*
```

En este archivo como se puede observar hay un ligero cambio de permisos. El archivo "prueba" ya que comienza mediante un "-" tiene permisos de lectura y ejecución para su dueño "core", de ejecución para los de su grupo "users", y de lectura y escritura para el resto de usuarios. Como habrá podido ver, al no haber permisos de escritura para su dueño, ni permisos de lectura y escritura para su grupo, ni permisos de ejecución para el resto de los usuarios, estos son representados mediante un "-".

Ahora que ya he explicado las características globales de todo directorio y archivo unix daremos paso a la explicación de las dos formas citadas anteriormente para establecer permisos en Linux

Forma absoluta:

La principal característica que tiene esta nomenclatura es la de asignar los permisos mediante valores octales. La base octal, o base 8, puede contener de los números 0 al 7. Es por ello, que tan solo son válidos estos números a la hora de asignar permisos. Fíjese en el siguiente cuadro:

Fig 1. Permiso Valor_Octal

```
-----
r | lectura 4
w | escritura 2
```

x | ejecucion 1

Claramente se observa que número octal corresponde a cada característica de un archivo o directorio en un sistema UNIX. Como mencione antes el comando utilizado por los Sistemas UNIX para establecer permisos recibe el nombre de `chmod`. A la hora de dar permisos en forma absoluta hemos de seguir la siguiente Sintaxis: "`chmod XYZ archivo`".

X representa al dueño del archivo, Y al grupo y Z al resto de usuarios, mientras que "archivo" es el nombre del archivo a especificar los permisos. Siempre que queramos atribuir mas de un permiso los numeros octales se sumarán. Observar los siguientes ejemplos para un mayor entendimiento:

```
chmod 460 archivo
chmod 755 archivo
chmod 050 archivo
chmod 000 archivo
chmod 777 archivo
```

EL primer ejemplo daria permisos de lectura al propietario (4), lectura y escritura al grupo (4+2), y ningún permiso al resto de usuarios (0).

El segundo ejemplo daría permisos de escritura, lectura y ejecución al propietario (4+2+1), permisos de lectura y ejecución tanto al grupo como al resto de usuarios (4+1)

El tercer ejemplo no daría ningún permiso al propietario (0), permisos de ejecución y lectura al grupo (4+1), y ningún permiso al resto de usuarios (0)

El cuarto ejemplo no tendría ningún permiso para nadie (0). Y finalmente el último ejemplo tendria todos los permisos para todos los usuarios (4+2+1)

Forma relativa:

A diferencia de la forma absoluta, la relativa no utiliza números octales o en base 8 para establecer los diferentes permisos, sino que se basa en una nomenclatura de letras. Al igual que en la forma absoluta, en la relativa también nos valemos del comando "`chmod`" para asignar los permisos.

Observe los siguientes cuadros:

Fig 1.

Letra

- a**, Engloba todos los usuarios, grupos y demas usuarios.
- g**, Engloba el grupo del propietario.
- o**, Engloba todos los demas usuarios no mencionados antes.
- u**, Engloba al usuario que creo dicho archivo

Fig 2.

Operador

- + Agrega la modalidad
- Elimina la modalidad
- = Elimina los permisos existentes y agrega los establecidos que sentencia que indiquemos.

Fig 3.

Permiso

- x**, Establecemos la ejecución
- r**, Establecemos la lectura
- w**, Establecemos la escritura

Siempre que se quiera atribuir permisos a un archivo/directorio en forma relativa se debe seguir el orden de los tres cuadros expuestos anteriormente. Por lo tanto primero se indicará a la persona o personas a la que se quiere atribuir dichos permisos, ejemplo:

chmod a+r archivo

Si se ha fijado en las Figuras anteriores, no te debería costar demasiado entender los permisos que atribuye la orden anterior. Primeramente como he dicho, se debe indicar al usuario/usuarios, en este caso esta la letra "a" que significa que se establecen permisos al dueño, grupo y demas usuarios. El siguiente simbolo que le sigue, indicará si se quiere agregar o eliminar permisos del archivo en cuestión, en este caso el simbolo "+" indica que deseo agregar permisos al archivo. Finalmente se indicarán los permisos a agregar. En este caso se indica "r" (lectura) para todos los usuarios "a".

Si por el contrario se quiere establecer varios permisos a las diferentes categorias que comprenden los sistemas UNIX se pueden utilizar las comas.
Ejemplo:

chmod o-wr, g-wr archivo

Que simplemente esta sentencia eliminaría ambos permisos, de escritura y lectura tanto para el grupo, como para el resto de usuarios o general.

[Indice General](#)

Links varios

La mejor que he encontrado es <http://sunsite.unc.edu/pub/Linux/> Incluye una extensa lista de las Linux Frequently Asked Questions list (FAQ)

Organizacion GNU/Linux <http://www.gnu.org/>

Organizacion mundial de Linux <http://www.linux.org/>

Sitio oficial de la version RedHat de Linux <http://www.redhat.com/>

Sitio oficial de la version Caldera de Linux <http://www.caldera.com/>

Sitio oficial de la version Suse de Linux <http://www.suse.de/>

Sitio oficial de la version Slackware de Linux <http://www.slackware.com/>

Sitio oficial de la version Debian de Linux <http://www.debian.org/>

Sitio oficial del servidor web Apache <http://www.apache.org/>

Documentacion oficial de Apache <http://httpd.apache.org/docs/>

Washington University FTP server "Sitio Oficial" <http://www.wu-ftp.org/>

Entorno grafico Gnome <http://gnome.org/>

Entorno grafico KDE <http://www.kde.org/>

[Indice General](#)