

Sistema Operativo Linux

Administración de servicios en red

M. en C. ALEJANDRO SOTO RAMOS
ESCOM

Historia De LINUX

La historia de LINUX inicia con el desarrollo de un pequeño programa llamado Minix, un sistema operativo tutorial, escrito por el científico Andrew Tannebaum, este sistema operativo gano popularidad siendo desarrollado para poder trabajar en diferentes plataformas de equipo de cómputo lo que inspiro al desarrollo de LINUX.

En la década de los 60's cuando todo el software se mantenía en una arquitectura cerrada dictada por el diseño de cada productor de hardware los Laboratorios Bell de AT&T en conjunto con GE trabajaron en el MIT en un proyecto llamado MULTICS (Servicio de computo de información Multiplexada) un antiguo sistema de tiempo compartido el cual tenia como finalidad proporcionar un soporte para diferentes arquitecturas de hardware compartiendo los recursos de hardware y software y así poder eliminar el concepto de arquitectura cerrada

En el año de 1969 el proyecto MULTICS fracasa por su alta demanda de espacio requerido en disco duro y memoria RAM, lo cual evita que el proyecto sea financiable, abandonándolo en este año.

Kenneth Thompson y Dennis Ritchie, ingenieros de laboratorios BELL deciden continuar con el proyecto por su cuenta, desarrollando herramientas de programación que les permitieron tener un mejor desempeño de los programas diseñados por ellos mismos.

Kenneth Thompson desarrolla un lenguaje de programación de bajo nivel al cual le llamó lenguaje B, éste es analizado y mejorado por Dennis Ritchie, desarrollando un lenguaje de programación más potente y robusto al cual le llamaron C, el cual se ha mantenido desde entonces hasta la fecha como un lenguaje de programación de excelencia para los diseñadores de programas que tienen un alto nivel de interacción con el hardware.

Para el año de 1971 es lanzada la segunda versión de UNIX, esta versión se diseñó en lenguaje C, lo que le permitió la portabilidad entre las diferentes arquitecturas de hardware, de este modo se inicia el diseño de la arquitectura abierta en los equipos de cómputo.

En el año de 1983 Novell compra UNIX a AT&T y lo registra con el nombre de System V obteniendo todos los derechos sobre el producto.

El nombre de UNIX se desprende de una aberración de Multics, ya que unix inicio como un proyecto menos ambicioso el cual sólo podía ser usado por una persona, donde cada parte del sistema estaba diseñada para hacer solo una cosa y hacerla bien, de aquí el nombre; “uni” que quiere decir “mono o uno” seguido de la letra X homófona

Introducción a GNU/Linux

UNIX surgió en 1969 en los Laboratorios Bell (Ken Thomson, Dennis Ritchie)

Dos grandes vertientes

BSD: SunOS, NetBSD, OpenBSD, Mac OS










System V: Solaris, Iris, Aix, Linux (año 1991)

Distribuciones Linux

- Slackware
- Gentoo
- Suse
- RedHat y derivados: Fedora, Mandriva (Mandrake)
- Debian y derivados: Ubuntu, knoppix, GnuLiNex

Posteriormente en 1990 un estudiante Finlandés inicio un proyecto personal basado en Minix, un sistema operativo tipo UNIX, su intención era desarrollar un sistema compatible con plataforma PC el cual fuera más confiable que Minix, comenzó creando drivers para dispositivos de hardware en lenguaje C mejorando su plataforma, para 1991 libera la primera versión de Linux sin ser una distribución oficial, la 0.01


Características De LINUX


-  Es un sistema operativo de tiempo compartido
-  Multiusuario
-  Multitarea
-  Multiproceso
-  Multiplataforma
-  Modular
-  Portable entre sistemas abiertos
-  Programable
-  Es un sistema tipo cliente - servidor

Filosofía De LINUX

 Su forma modular lleva a su filosofía Principal

“Entre más pequeño mejor”.

 Permite la reducción de esfuerzo combinando las herramientas.

 Es un sistema personalizable de acuerdo a las necesidades de cada usuario.

Plataformas y Distribuciones Más Comunes

El tipo de plataforma se refiere a la arquitectura del hardware sobre el que va a correr nuestro linux, para ello se debe conocer las características de nuestro hardware para adquirir el tipo de distribución apropiada de acuerdo al equipo donde se va a instalar.

Las plataformas principales son:

✍️ I386 Para equipos tipo PC compatibles.

✍️ X64 Para equipos tipo PC de 64 bits

✍️ PPC Para equipos Maquintosh

✍️ Sparc Para estaciones de trabajo

✍️ Alpha Para arquitectura de trabajo con procesadores en paralelo de tipo alpha.

Existen muchas distribuciones que pueden ser adquiridas por medio de Internet, haciendo la compra en el sitio oficial que le distribuye o descargándolo gratuitamente de los sitios de FTP, todas las distribución utilizan la versión del kernel que se actualiza constantemente, la diferencia radica en que los diseñadores y programadores le dan su toque especial, principalmente en presentación de la interfaz gráfica, implementación de nuevas herramientas, programas, formatos gráficos, etc.

Las distribuciones más comunes son:



<http://www.slackware.com/>



<http://www.redhat.com/>



<http://www.debian.org/>



<http://www.linux-mandrake.com/>



<http://www.suse.com/>



<http://www.freebsd.org/>



<http://www.netbsd.org/>



<http://www.turbolinux.com/>



<http://www.knopper.net/>

Componentes Del Sistema Operativo LINUX


LINUX esta compuesto básicamente de cuatro capas, la capa mas interna esta conformada por el hardware, que es el conjunto de piezas físicas del equipo de cómputo.


La segunda capa es el Kernel o núcleo del sistema, su función principal es interpretar las instrucciones proporcionadas por el usuario y convertirlas en lenguaje de máquina e indicarle al hardware lo que tiene que realizar con dicha información.


Como interpretar los números de las versiones:

Las versiones del kernel se numeran con 3 números, de la siguiente forma: XX.YY.ZZ

XX: Indica la serie principal del kernel. Hasta el momento solo existen la 1 y 2. Este numero cambia cuando la manera de funcionamiento del kernel ha sufrido un cambio muy importante. (La versión 3 es del festejo por el aniversario)

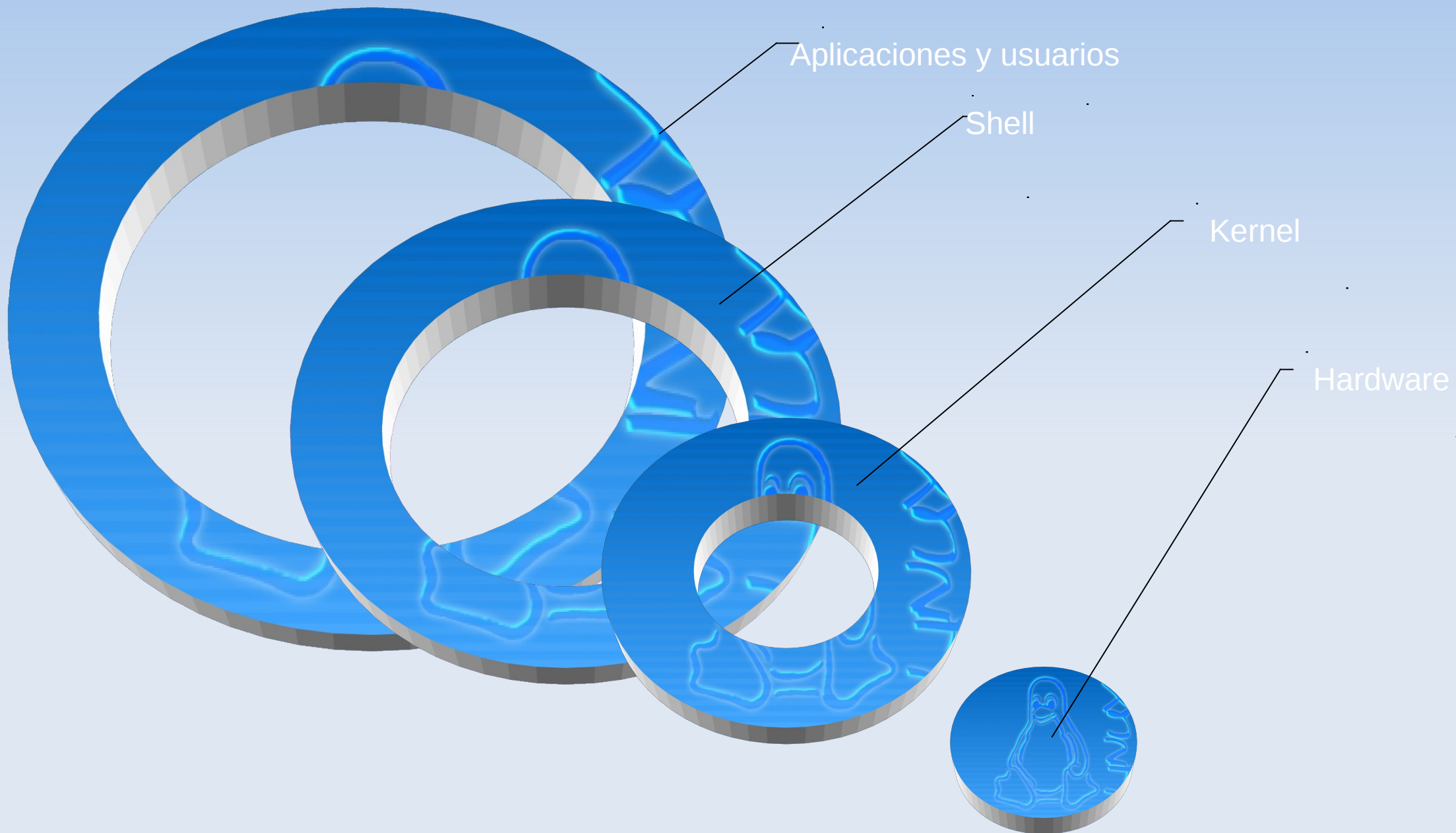
 **YY:** Indica si la versión es de desarrollo o de producción. Un numero impar, significa que es de desarrollo, uno par, que es de producción.

 **ZZ:** Indica nuevas revisiones dentro de una versión, en las que lo único que se ha modificado, son fallos de programación/bugs.

 **El número D (Cuatro números a partir de la 2.6) la nueva política de versiones.”Bug-fixes y parches de seguridad”.**

La tercer capa esta conformada por el grupo de los Shells o interpretes de comandos, los cuales funcionan como la interfaz entre el usuario y el kernel, proporcionan las herramientas para que el usuario se pueda comunicar con el núcleo del sistema de LINUX.

La cuarta y última capa es donde se encuentra el usuario junto con los programas y aplicaciones que se le han agregado al sistema como hojas de cálculo, lenguajes de programación, manejadores de bases de datos, procesadores de texto, etc.



Linux

- Hoy en día existen muy buenas razones para optar por Linux ya que, el sistema ofrece estabilidad, seguridad y velocidad. Otro aspecto importante es su capacidad de conectividad en redes que ha sido decisiva para la conquista del mercado de servidores. Los gurús del Linux aprecian la disponibilidad del código fuente lo que proporciona al sistema operativo un alto nivel de independencia y flexibilidad.

GNU/Linux

- GNU/Linux es un sistema operativo. Esto significa que es un conjunto de instrucciones que nos va a permitir "hacer cosas", cualquier tipo de tarea, con nuestro equipo. Sus características más notables son:
- Proviene de UNIX, otro sistema operativo, y es casi un derivado de él.
- Es parte del proyecto GNU, lo que significa, entre otras muchas cosas, que es libre y que muchas veces no estás obligado a pagar por él. Puedes usarlo de modo gratuito y además puedes modificar su código fuente, para adaptarlo a tus propias necesidades o para contribuir en su continuo desarrollo, en el que toman parte programadores de todo el mundo (<http://www.gnu.org>).

GNU/Linux

- Es potente, seguro y estable. Debido a ello resulta un sistema operativo ideal para servidores ya que cumple muy bien esta función aunque como sistema operativo de oficina, de escritorio o de publicación también es excelente.
- Existen infinidad de programas para cualquier tipo de tarea que se quiera desarrollar y aunque muchos de ellos son gratuitos (no por ello de menos calidad que los comerciales, pues muchas veces los superan), también existen aplicaciones comerciales.

GNU/Linux

- Es un sistema multiusuario real y multitarea y funciona de manera muy productiva en redes. Desde sus raíces UNIX siempre lo ha sido. En definitiva Linux pone todo lo bueno de los grandes UNIX y algunas cosas más al alcance de todo el mundo.

GNU/Linux

- Puede decirse que Linus Trovalds creó este sistema operativo en 1991. Escribió un KERNEL (parte principal de un sistema operativo) y lo dejó a disposición de sus amigos y luego de toda la Internet para que cualquiera pudiese mejorarlo.
- Siempre se ha seguido la regla POSIX para que el sistema operativo sea compatible con otros UNIX y de esta manera tener mayor software a su disposición, y que el que se cree para Linux valga también en otros UNIX. Algún tiempo más tarde, para adaptar el sistema operativo y facilitar la instalación, nacieron las compañías distribuidoras de Linux.

GNU/Linux

- Entre ellas destacan RedHat, Debian y SuSE. Estas distribuidoras de software pueden cobrar por ofrecer sus productos en DVD o en algún otro soporte, pero deben poner su código a disposición del público, si han utilizado software GNU (bajo licencia GPL, que dice entre otras cosas que si usas código GPL en tu programa, tu programa se convierte automáticamente en GPL).
- Actualmente Linus Torvalds sigue coordinando el trabajo de los programadores de todo el mundo en el núcleo del sistema operativo para mejorarlo y adaptarlo al nuevo hardware.

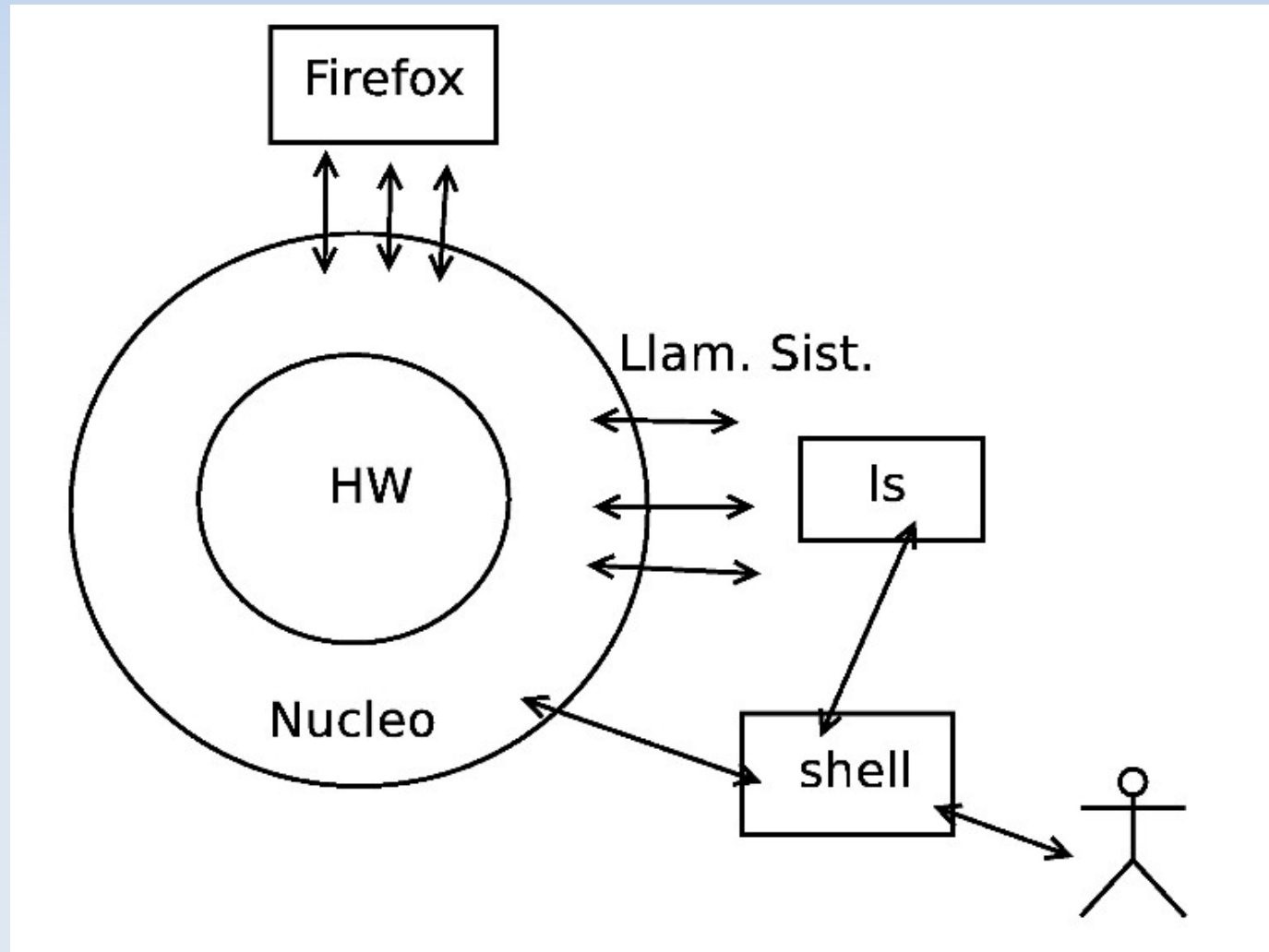
El arranque de Linux

- Cuando la BIOS termina su chequeo, empieza a buscar dónde hay software para ser ejecutado. Si los CDRoms no son de arranque, pasa al primer disco duro por defecto. En el principio del disco duro o MBR en una partición queda instalado con Linux un cargador de arranque.
- Los dos más usados son LILO (Linux LOader) y GRUB. Nada más al arrancar se ejecuta este cargador cuya misión permitir al usuario elegir sistema operativo de los que tenemos en nuestro equipo instalados
- Seleccionamos nuestra versión de Linux y el cargador da paso al Kernel de Linux que empieza a hacer sus chequeos y a montar (hacer utilizable) el sistema de archivos.

El arranque de Linux

- Kernel (Núcleo): elemento más importante. Permite que las aplicaciones accedan al hardware. Es responsable de la gestión de recursos, seguridad, etc
- Procesos de usuario: distintos programas ejecutándose concurrentemente en un sistema
- La interacción entre el núcleo y los procesos se hace mediante llamadas al sistema (system calls)
- La shell es un interfaz de usuario en modo texto. Es una aplicación como otra cualquiera

El kernel de Linux



El kernel de Linux

La Free Software Foundation (Richard Stallman) considera que:

Linux es estrictamente el kernel

Los procesos de usuario (programas y otras utilidades básicas para el sistema) provienen del proyecto GNU (y algunos otros).

Al conjunto se le debe llamar GNU/Linux.

Un número importante de personas y organismos se oponen a esta definición. La mayor' de la gente lo llama simplemente
Linux

Linux

Hay software libre para cualquier S.O.

Hay software propietario para Linux

Cuatro libertades. Quien lo recibe tiene:

libertad de uso. Usarlo como quiera, donde quiera

libertad de redistribución. Redistribuirlo a quien quiera, como quiera

libertad de modificación. Modificar, adaptar, corregir, mejorar

libertad de distribuir las modificaciones

Imprescindible: disponibilidad de código fuente.

El sistema de archivos

- El sistema de archivos es más o menos "la forma de escribir los datos en el disco duro". El sistema de archivos nativo de Linux es el EXT2. Ahora proliferan otros sistemas de archivos con journalising (si se arranca sin haber cerrado el sistema, no necesitan hacer un chequeo sino que recuperan automáticamente su último estado), los más conocidos son EXT3, EXT4 ReiserFS y XFS.

El sistema de archivos

- Linux soporta varios tipos de sistemas de ficheros. Entre los más importantes podemos destacar los siguientes:
MINIX: el más antiguo, presume de ser el más seguro, pero es bastante limitado en las características que proporciona. Un sistema de ficheros de este tipo solo puede tener 64 MB.
- EXT2: es el sistema de ficheros nativo de Linux. Está diseñado para ser compatible con versiones anteriores, así que las nuevas versiones del código del sistema de ficheros no requerirán rehacer los sistemas de ficheros existentes.
- EXT3: es una modificación del ext2 para añadirle funcionalidades de journaling.

El sistema de archivos

- VFAT: este tipo permite utilizar sistemas de ficheros de Windows (FAT, FAT32), y actualmente está soportado el sistema de ficheros de Windows NT, pero solo fiable en solo lectura.
- Iso9660: es el sistema de ficheros estándar para CD-ROM.
- NFS: un sistema de ficheros en red que permite
- XT2: es el sistema de ficheros nativo de Linux. Está diseñado para ser compatible con versiones anteriores, así que las nuevas versiones del código del sistema de ficheros no requerirán rehacer los sistemas de ficheros existentes.
- HPFS: es el tipo de sistema de ficheros de OS/2
- SYSV: es el tipo de sistema de ficheros de SystemV/386, Coherent y Xenix.

El sistema de archivos “ESTRUCTURA DE DIRECTORIOS”

El sistema de archivos

- El sistema de archivos /proc permite acceder fácilmente a ciertas estructuras de datos del kernel, como es la lista de procesos. Convierte estas estructuras de datos en algo parecido a un sistema de archivos y por tanto da la posibilidad de manipularlas con las herramientas habituales de manipulación de archivos. Hay que tener en cuenta que aunque se le denomine sistema de archivos, ninguna parte del sistema de archivos /proc toca el disco

El sistema de archivos

- Recursos principales por los que “luchan” los procesos. Linux soporta múltiples procesadores.
- Dispone de memoria virtual
- El administrador debe conocer la utilización que se está haciendo de estos recursos.
- CPU podemos ver su utilización con top, uptime.
- La memoria con free, **top**, vmstat.

El sistema de archivos

- En un sistema Unix todo son archivos: memoria física, ratón, modems, teclado...
- Filosofía de diseño de gran éxito y potencia, aunque también peligrosa: un simple error de permisos puede permitir modificar todo un disco duro.
- Tipos de archivos:
 - Archivos texto planos
 - directorios
 - ficheros especiales (dispositivos)
 - orientados a carácter
 - orientados a bloque

La estructura de directorios

- La estructura de directorios que sigue Linux es parecida a la de cualquier UNIX. No tenemos una "unidad" para cada unidad física de disco o partición como en Windows, sino que todos los discos duros o de red se montan bajo un sistema de directorios en árbol, y algunos de esos directorios enlazan con estas unidades físicas de disco.
- MUY IMPORTANTE: Las barras en Linux al igual que en cualquier UNIX son inclinadas hacia la derecha

La estructura de directorios

Directorio	Descripción
/	Es la raíz del sistema de directorios. Aquí se monta la partición principal Linux EXT.
/etc	Contiene los archivos de configuración de la mayoría de los programas.
/home	Contiene los archivos personales de los usuarios.
/bin	Contiene comandos básicos y muchos programas.
/dev	Contiene archivos simbólicos que representan partes del hardware, tales como discos duros, memoria...

La estructura de directorios

- `/mnt` Contiene subdirectorios donde se montan (se enlaza con) otras particiones de disco duro, CDRoms, etc.
- `/tmp` Ficheros temporales o de recursos de programas.
- `/usr` Programas y librerías instalados con la distribución
- `/usr/local` Programas y librerías instalados por el administrador
- `/sbin` Comandos administrativos

La estructura de directorios

- `/lib` Librerías varias y módulos ("trozos") del kernel
- `/var` Datos varios como archivos de log (registro de actividad) de programas, bases de datos, contenidos del servidor web, copias de seguridad...
- `/proc` Información temporal sobre los procesos del sistema (explicaremos esto más en profundidad posteriormente).

La estructura de directorios

- Cualquier sistema de archivos lo montaremos sobre el árbol de directorios, con el comando mount. Para quitarlo con umount.
- Para tener información de los sistemas tenemos los comandos df y du.
- Podemos establecer cuotas (quotaon) del sistema a los usuarios, con edquota, verlas con quotacheck.
Dispone de premisos para ficheros unix y acl.

La estructura de directorios

Tener todo el sistema en una sola partición es mala idea ya que no es flexible y hace difícil de administrar el servidor.

Lo más recomendable es hacer particiones pensando en la carga de archivos del sistema, de los usuarios, servicios y otras consideraciones, logrando con ello facilitar las tareas de respaldo, actualización, administración fácil y eficiente, entre muchas otras ventajas

La estructura de directorios

Dispositivo (device) es el dispositivo al que está conectado el mouse. Las posibilidades son las siguientes.

- `/dev/psaux`, si se trata de un mouse PS/2.

- `/dev/ttyS0`, si está conectado al COM1.

- `/dev/ttyS1`, si está conectado al COM2.

- `/dev/ttyS2`, si está conectado al COM3.

- `/dev/ttyS3`, si está conectado al COM4.

- `/dev/usb/mouse`, si es un mouse USB.

- `/dev/mouse`, que normalmente es un symlink al dispositivo correspondiente al mouse.

La estructura de directorios

En UNIX, la mayoría de los dispositivos físicos están representados en el sistema de archivos por elementos especiales de tipo dispositivo (device). Estos archivos se agrupan dentro del directorio /dev. Veamos un ejemplo:

```
$ cd /dev
$ ls -l ttyS0      fd0 hda hda1 dsp
crw-rw----    1 root      audio      14,   3  abr  22 2000    dsp
brw-rw----    1 root      floppy      2,   0  abr  22 2000    fd0
brw-rw----    1 root      disk        3,   0  abr  22 2000    hda
brw-rw----    1 root      disk        3,   1  abr  22 2000    hda1
crw-rw----    1 root      dialout     4,  64  ene   9 12:38    ttyS0
```

Si se necesitan nuevos archivos de dispositivo, el script /dev/MAKEDEV puede ejecutarse con los argumentos adecuados para crear los dispositivos o grupos de dispositivos que se necesiten

La estructura de directorios

Las tuberías son ficheros especiales que no almacenan información en el disco, sino que la guardan en memoria en espera de que algún programa la lea. En este sentido funcionan como un buffer tipo FIFO (First In First Out). Permiten conectar la salida de un programa con la entrada de otro, de forma parecida a como lo hace el shell con el operador "|". Las tuberías se crean con el comando `mkfifo(1)`.

primero creamos una tubería llamada `mififo`

```
$ mkfifo mififo
```

```
$ ls -l
```

Ahora ejecutamos un proceso que escriba en la tubería. El proceso quedará bloqueado en espera de que alguien lea de la tubería, por eso lo ejecutamos en el background:

```
$ ls > mififo &
```

```
$ ls -l
```

Ahora leemos de la tubería. Esto hace que el proceso `ls` pueda escribir los datos y termine:

```
$ cat mififo
```

```
mififo|
```

La estructura de directorios /etc

Almacena los parámetros de configuración de distintos elementos del sistema

Algunos ejemplos

clock

ide

keyboard

pcmcia

sound

ssh

La estructura de directorios /etc

Las bases de datos de los usuarios y passwords: passwd, shadow, group, gshadow

El archivo de configuración de init, que es el padre de todos los procesos:

inittab

El directorio de los scripts de arranque de servicios: /etc/init.d (junto con los links simbólicos hacia ellos, que están en /etc/rc*.d/) El directorio en donde se configuran variables por defecto para algunos de los servicios que arranca init: /etc/default/

Configuración esencial de filesystems: fstab

Configuración esencial de red: hostname, hosts, network/interfaces

Variables de entorno por defecto para shells de usuarios: environment y profile

Nombres de archivos en Linux

Los nombres de archivos en Linux (como en todos los UNIX) distinguen mayúsculas de minúsculas, esto es, son "case sensitive". Los archivos README, readme, REadme y rEadme por ejemplo son archivos distintos y por lo tanto al ser nombres distintos pueden estar en el mismo directorio.

En Linux los archivos no tienen por qué tener una extensión. La suelen tener a modo orientativo, pero no es en absoluto necesario. Linux sabe qué contiene cada archivo independientemente de cuál sea su extensión. Por comodidad, podremos llamar a todos nuestros archivos de texto con la extensión .txt, o .doc

Nombres de archivos en Linux

Los nombres de archivos o directorios pueden ser muy largos, de más de 200 caracteres, lo cual nos da bastante flexibilidad para asociar el nombre de un archivo a lo que contiene. No obstante, hay ciertos caracteres que nunca se deberían utilizar a la hora de nombrar un archivo.

El espacio, tampoco son recomendados otros caracteres raros como signos de puntuación (a excepción del punto), acentos o la ñ. En algunos casos Linux ni siquiera nos permitirá usarlos.

Utilizar letras A-Z, a-z, los números (0-9), el punto, el guión (-) y el guión bajo (_)

Espacio en disco

df

df (Disk Free) nos informa del espacio disponible de las unidades de disco (locales o de red) que tengamos montadas.

df

S.ficheros	1K-blocks	Used	Available	Use%	Montado en
/dev/hdd1	6040288	3247924	2485528	57%	/
/dev/hda2	11719052	3526412	8192640	31%	/home
none	128008	0	128008	0%	/dev/shm

Una opción útil de la que dispone es -m, que mostrará el espacio en MegaBytes.

Montar dispositivos

En realidad, las operaciones de montaje y desmontado las llevan a cabo los comandos `mount` y `umount`, por ejemplo:

```
# mount /dev/hdc /cdrom
```

Antes del montaje, el directorio `/cdrom` debe estar creado. Tras el montaje, el contenido del dispositivo montado “aparece” en `/cdrom` como si se tratara de un directorio más del sistema. De la misma forma se haría para un diskette o una usb:

```
# mount /dev/fd0 /floppy
```

Montar dispositivos

Disco duro IDE y CDROM: /dev/hdXY

X: Número de disco/dispositivo_IDE (a, b, c...)

Y: Número de partición (1, 2, 3...)

/dev/hda

/dev/hda1

/dev/hdb3

Disco duro SCSI, SATA y externos USB: /dev/sdXY

/dev/sda

/dev/sda1

/dev/sdc4

Montar dispositivos

Linux reconoce muchos formatos posibles de discos, diskettes y CD-ROM. En la mayoría de los casos el sistema identifica automáticamente el formato del dispositivo, pero cuando esto no es posible, podemos indicar el formato con la opción `-t` de `mount`, por ejemplo:

```
# mount -t vfat /dev/fd0 /floppy
```

montaría un diskette con formato MS-DOS y soporte para nombres largos. Otros formatos comunes son: `msdos` (formato MS-DOS normal), `iso9660` (formato de los CD-ROM's) y `ext2` (formato nativo de Linux)

Montar usb

creas un sitio donde montar la particion

```
mkdir /media/discoUSB
```

luego lo montas como root

```
mount /dev/sda1 -t vfat /media/discoUSB
```

para que no tengas que ser root, escribe (como root) esto
en /etc/fstab

```
/dev/sda1 /mnt/discoUSB vfat users,noauto,umask=000,exec 0 0
```

```
mount /media/discoUSB
```

Archivo /etc/fstab

Para facilitar el uso de mount y para indicar al sistema que dispositivos forman el sistema de archivos, existe el archivo /etc/fstab. En él se indican que dispositivos han de montarse en qué lugar y con que opciones

Para crear particiones se utiliza el comando fdisk o cfdisk:

```
# cfdisk /dev/hda
```

- . Los cambios propuestos no se hacen efectivos hasta que se escribe la tabla de particiones.
- Si el sistema esta funcionando se debe conocer previamente que particiones estan montadas y activas (con el comando mount o df) para no modificarlas pues se dañaría el sistema.
- Una vez escrita la tabla de particiones, se ha asignado un tipo de sistema de fichero a cada una de las particiones establecidas en dicha tabla pero no se les ha dado el formato adecuado.

Fichero /etc/fstab

/dev/hda5 Partición o dispositivo

/respado Punto de montaje en el sistema

ext3 Tipo de sistema

defaults Opciones de montaje asociadas al dispositivo

0 Sistemas que requieren dump

0 Orden en que son revisados los sistemas.

Formateando nuevas particiones

En la particion de disco duro libre denominada ext3, se realiza un proceso de division por bloques de tamaño entorno a 4KB numerandolos secuencialmente. Estos bloques se agrupan como minimo, 6 grupos distintos:

- Grupo 1: Bloque 1: Bloque donde se coloca el sector de arranque
- Grupo 2: Bloque 2: Se especifica el tamaño de los restantes bloques
- Grupo 3: Bloque 3 hasta n1: Se especifica el mapeado de los i-nodos
- Grupo 4: Bloque n1+1 hasta n2: Se especifica el mapeado de los datos
- Grupo 5: Bloque n2+1 hasta n3: Espacio de i-nodos
- Grupo 6: Bloque n3+1 hasta n4: Espacio de datos

Para formatear una particion en Linux se utiliza el comando mkfs

```
# mkfs -t ext3 /dev/hda3
```

Formateando nuevas particiones

Si existen herramientas de reparación del sistema de archivos que, incluso, se ejecutan automáticamente cada vez que al arrancar el sistema se detecta que en la última sesión no fueron cerrados apropiadamente.

El comando para reparar el sistema de ficheros es fsck o e2fsck en el caso de sistemas de ficheros ext3:

```
# e2fsck /dev/hda3
```

Es importante tener en cuenta que este proceso solo debe llevarse a cabo con la partición no montada o bien montada de solo lectura lo cual plantea un problema con la partición raíz del sistema.

Formateando nuevas particiones

Una vez formateada una nueva particion, por ejemplo, /dev/hda3, se puede acceder a ella montándola en algun directorio designado para contenerla:

```
# mount -t ext2 /dev/hda3 /mnt
```

Si se pretende disponer automaticamente de esta particion al arrancar el sistema se necesita añadir una linea al fichero /etc/fstab. Si por ejemplo se quisiera montar la partición /dev/hda3 en el directorio /home la línea sería la siguiente:

```
/dev/hda3          /home      ext2      defaults,errors=remount-ro 0 1
```

Formateando nuevas particiones

Crear una partición swap

El proceso comienza por crear una nueva particion con el comando cfdisk siguiendo el mismo procedimiento que contamos previamente. Lo unico que hay que tener en cuenta es que el tipo de la nueva particion debe ser Linux swap. Para formatear esta particion se utiliza el siguiente comando:

```
# mkswap /dev/hdxx
```

Para activarla:

```
# swapon /dev/hdxx
```

Formateando nuevas particiones

El proceso de creacion y activacion de un archivo swap es como sigue:

1. Se crea un archivo relleno de ceros de un tamaño adecuado:

```
# dd bs=1024 if=/dev/zero of=/dev/swapfile count=1000
```

Con esta orden se crea un fichero /dev/swapfile de tamaño 1024X1000 (aprox. 1M)

2. Se formatea el archivo swap:

```
# mkswap /dev/hdxx 1000
```

3. Activar el archivo swap:

```
# sync
```

```
# swapon -v /dev/swapfile
```

Formateando nuevas particiones

Actualización de sistemas ext2 a ext3

Como ext3 es sólo una actualización a ext2, los sistemas de archivos ext2 pueden convertirse de forma sencilla a ext3, sin tener que formatear particiones ni copiar discos

Añadimos primero un registro de transacciones a los sistemas de archivos. No hace falta desmontar los sistemas de archivos ni nada:

```
# tune2fs -j /dev/hdb1
```

```
# tune2fs -j /dev/hdb3
```

Luego alteramos /etc/fstab para reflejar el cambio en el sistema de archivos:

La próxima vez que arranquemos el sistema, los sistemas de archivos funcionarán como ext3. Podemos comprobarlo en los mensajes de arranque del kernel:

```
# dmesg | less
```

Usuarios en Linux

Características de usuarios Unix

Los sistemas Unix son sistemas multiusuario. Cada usuario tiene una serie de características propias y asociadas:

uid: identificativo de usuario (debe ser único)

gid: identificativo de grupo

home: carpeta de trabajo o personal

shell: interprete de comandos

Comandos:

who/whoami/id (muestra información de usuarios)

su/sudo (cambia de usuario o privilegios)

GESTIÓN DE USUARIOS Y CUENTAS EN LINUX

Elementos Básicos De Una Cuenta De Usuario

Una cuenta de correo esta basada en un login y un password, ésta se complementa con el nombre del servidor y el dominio del mismo en el cual se encuentra alojada dicha cuenta

El login es el identificador del usuario dentro del servidor, dicho login es único, es decir, no puede haber dos cuentas en el servidor con el mismo login ya que éste es el nombre con que se conoce al usuario dentro del servidor.

usuario@equipo:

Password

El password Es la clave de acceso al sistema, sólo debe conocerla el dueño de la cuenta ya que permite el acceso al servidor, a todos sus servicios disponibles y a la información del usuario.

Cuando el sistema nos pide la contraseña y nosotros la tecleamos, ésta no es impresa en el monitor por seguridad para evitar que sea vista por otra persona.

Cambio De Contraseña

passwd

Éste comando nos permite realizar el cambio de contraseña de una cuenta, para poder realizar el cambio necesitamos saber la contraseña actual de la cuenta ya que por seguridad el sistema nos pedirá que insertemos dicha contraseña para a continuación teclear la nueva contraseña dos veces para verificar que haya sido escrita correctamente.

Salida Del Sistema

Es necesario indicarle al sistema que queremos terminar la sesión de trabajo en el servidor, para ello existen varios comandos, entre ellos se encuentran los comandos `exit` y `logout` o la combinación de teclas `Ctrl+d`.

Ambos comandos nos permiten terminar la sesión en el servidor, al momento que lo tecleamos y presionamos enter se cierra la conexión en el servidor.

Usuarios en Linux

Gestión de Usuarios

- adduser (crear usuarios)
- usermod (modificar usuarios)
- deluser (eliminar usuarios)

Gestión de Grupos

- addgroup (añade grupo o usuario a grupo)
- groupmod (modifica grupo)
- delgroup (elimina grupo o usuario de grupo)

Usuarios en Linux

/etc/passwd

Contiene lista de usuarios reconocidos por el sistema,
consultado por el sistema en un login

login (nombre)

password encriptado (puede estar en archivo shadow)

UID

GID por defecto

GECOS

directorio home

shell por defecto

Usuarios en Linux

`/etc/passwd`: Archivo donde se almacena toda la información relativa a los usuarios del sistema, excepto la contraseña (encriptada) si se utilizan shadow passwords. Se recomienda utilizar shadow passwords por cuestiones de seguridad (evitar el “ataque con diccionario”).

`/etc/shadow`: En un sistema con el mecanismo de shadow passwords activado es el archivo que almacena las contraseñas (encriptadas) de los usuarios.

`/etc/group`: Fichero donde se guardan los nombres y los identificadores numericos de los grupos además de la lista de usuarios adscritos a cada grupo.

`/etc/gshadow`: Fichero que guarda información complementaria para grupos si se activa el mecanismo de shadow passwords.

Usuarios en Linux

Nombres (login)

Hasta 32 caracteres (todos excepto “:” y LF)

Recomendado límite de 8. Viejos UNIX limitan a 8 (adoptado por NIS)

Se recomienda usar reglas para asignar cuentas (nombres se usan en email!)

/etc/mail/aliases puede ser usado para equivalencias entre email y login

Password encriptado

Se usa DES (usa 8 caracteres) o MD5

Cuando se crea una cuenta, debe asignársele un password usando comando `passwd`

Un * en /etc/passwd (o /etc/shadow) significa que no se puede hacer login.

Vacío significa que no requiere password!!

Usuarios en Linux

UID y GID

UID de 32 bits (0 reservado para root)

Pseudo-usuarios (bin, daemon, etc.) se les asigna los números bajos

Usuarios normales: recomendado $\text{UID} > 100$

Conveniente usar mismo UID para el mismo usuario en la red (NFS!!)

Los grupos están definidos en `/etc/group`

Un usuario puede estar hasta en 32 grupos

Usuarios en Linux

GECOS

General Electric Comprehensive Operating System.

Usado para mandar login en trabajos batch de Unix a mainframe en Bell

Puede estar vacío

Se usa para información personal (nombre, telefono, oficina, etc.)

El comando finger interpreta el campo como nombre, oficina, etc.

El comando chfn permite modificarlo

Usuarios el Linux

Fichero oculto que incluye la codificación y las restricciones de las claves de acceso a las cuentas. Sus campos son:

Usuario:clave:F1:N1:N2:N3:N4:Caduc:

1. Nombre de usuario.
2. Clave codificada.
3. Fecha del último cambio de clave.
4. Días hasta que la clave pueda ser cambiada.
5. Días para pedir otro cambio de clave.
6. Días para avisar del cambio de la clave.
7. Días para deshabilitar la cuenta tras su caducidad.
8. Fecha de caducidad.
9. Reservado (ignorado por Red Hat).

Nota: Las fechas se expresan como el no de días desde el 1/1/1.970.

Gestion de usuarios

Cambiando el propietario y grupo

Para cambiar el propietario o grupo puede hacerse a través del nombre o del identificativo (uid o gid).

- chown (cambiar el propietario)
- chmod (cambiar el grupo)

Gestion de usuarios

```
linux:~$ adduser alumno
```

```
linux:~$ addgroup consultec
```

```
linux:~$ adduser --home /home/consultec/alumno --shell  
/bin/sh --uid 5001 --ingroup consultec alumno
```

```
linux:~$ usermod --shell /bin/bash alumno
```

```
linux:~$ adduser alumno softwarelibre
```

- Esto añade a “alumno” al grupo “softwarelibre”

Gestion de usuarios

Useradd es el comando para agregar usuarios genérico para todos los linux, es más complejo y se requiere conocimiento más profundo sobre administración de usuarios y grupos.

Se debe conocer los grupos que existen en el sistema, la secuencia de UID del sistema de usuarios entre otras cosas.

A parte de agregar los registros hay que crear el directorio del usuario, otorgarle privilegios de usuario y grupo y asignarle un password a la cuenta.

Gestion de usuarios

```
linux:~$ ls -l ~/documentos/linux.pdf
```

```
-rw-rw-r-- 1 alumno alumno 2684 ene  2 21:09 linux.pdf
```

```
linux:~$ chown alumno:consultec ~/documentos/linux.pdf
```

```
linux:~$ ls -l ~/documentos/linux.pdf
```

```
-rw-rw-r-- 1 alumno consultec 2684 ene  2 21:09 linux.pdf
```

```
linux:~$ id aktor
```

```
uid=500(aktor) gid=500(aktor) grupos=500(aktor)
```

```
linux:~$ chown 500 ~/documentos/linux.pdf
```

```
linux:~$ ls -l ~/documentos/linux.pdf
```

```
-rw-rw-r-- 1 aktor consultec 2684 ene  2 21:09 linux.pdf
```

Gestion de usuarios

Userdel Permite eliminar usuarios, aquí tenemos que utilizar la opción -r

para que haga un borrado recursivo de todo lo que pertenece al usuario a eliminar, de otra forma solo se eliminarán los registros del usuario de los archivos passwd, shadow y group, posteriormente el administrador tendrá que eliminar todo lo que pertenecía al usuario a mano.

Gestion de usuarios

Useradd es el comando para agregar usuarios genérico para todos los linux, es más complejo y se requiere conocimiento más profundo sobre administración de usuarios y grupos.

Se debe conocer los grupos que existen en el sistema, la secuencia de UID del sistema de usuarios entre otras cosas.

A parte de agregar los registros hay que crear el directorio del usuario, otorgarle privilegios de usuario y grupo y asignarle un password a la cuenta.

Gestion de usuarios

En ocasiones es necesario modificar el perfil de un usuario, para ello el sistema cuenta con el comando usermod, aunque tiene muchas opciones solo vamos a describir las más importantes.

usermod -u	modifica el UID del usuario
usermod -g	modifica el grupo principal de usuario
usermod -G	modifica los grupos adicionales del usuario
usermod -d	modifica el directorio HOME del usuario
usermod -s	modifica el shell por default
usermod -c	Modifica el comentario (campo de GECOS)
usermod -f	Asigna/modifica fecha de inactividad
usermod -e	modifica/asigna fecha de expiración de la cuenta
usermod -p	Cambia el password del usuario
usermod -L	Bloquea el acceso de la cuenta (Lock)
usermod -U	Desbloquea una cuenta bloqueada (Unlock)

Gestion de usuarios

Para agregar un grupo nuevo se usa el comando `groupadd`, todos los registros de los grupos se encuentran en el archivo `/etc/group`.

Algunos sistemas permiten agregar un grupo privado para cada usuario, este no es el caso de `debian`, ya que tiene un grupo llamado `users` al cual pertenecen todos los usuarios por default, aunque es posible modificar las propiedades para apropiarlo a las necesidades administrativas y funcionales del sistema.

Gestion de usuarios

Eliminación de un grupo

Antes de eliminar un grupo es conveniente revisar que ningún usuario tenga este grupo como principal, ya que el sistema no le permitirá eliminar el grupo lo que significa una reorganización del sistema.

Permisos

Todos los archivos en Unix tienen permisos que indican que y quien puede hacer o no hacer una acción con el archivo.

Es la base de la seguridad de Unix.

2 formas de notación:

- Modo alfabético
- Modo octal

Atributos de los archivos.

Tipos de archivos:

Dentro del sistema operativo vamos a encontrar archivos de diferentes tipos, los cuales los vamos a distinguir por el primer bit de sus propiedades.

El atributo del primer bit nos indica si el elemento listado es directorio o archivo, así mismo nos indica de que tipo de archivo se trata.

Archivos más comunes.

d Nos indica que el elemento es un directorio.

_ Archivo ordinario.

| El elemento listado es una liga suave.

c Archivo de tipo carácter (Archivos de acceso de hardware mediante un bit como el mouse).

b Archivo de bloque (acceso por bloques en paralelo como la impresora).

Permisos

Existen tres tipos de permisos que se aplican tanto a los archivos como a los directorios, ellos son:

r lectura

w escritura

x ejecución

Debemos asignar los permisos a los archivos de acuerdo a la función que van a realizar dependiendo para lo que fueron creados.

r El permiso de lectura permite que el archivo pueda ser leído o copiado.

w Permite escribir en el archivo, hacer modificaciones o borrarlo si así lo desea.

x Los archivos ejecutables son aquellos que pueden realizar un proceso.

Estos permisos se encuentran asignados en tres bloques, los cuales representan al usuario o dueño del elemento listado, al grupo al que pertenece el usuario y a los otros usuarios del sistema, de ello que los atributos se componen de 10 bits, el primer bit indica el tipo de elemento que esta listado (archivo o directorio), los siguientes 9 bits representan los tipos de permisos asignados a cada elemento listado.

Ejemplo de listado de permisos

```
dante% ls -l
total 426
-rw-r--r--    1 carlos  becas      30208 Sep 29 23:54 apa1.jpg
-rw-----    1 carlos  becas       1170 Sep 29 20:14 dead.letter
-rw-r--r--    1 carlos  becas     86796 Sep 29 23:53 Guerra1.jpg
drwxr-xr-x    2 carlos  becas       512 Sep 30 01:16 hola
lrwxrwxrwx    1 root    other         5 Sep 28 15:11 hola1 -> hola1
-rw-r--r--    2 carlos  becas         0 Sep 28 15:11 liga
-rw-r--r--    2 carlos  becas         0 Sep 28 15:11 liga1
-rw-----    1 carlos  becas     13167 Sep 29 23:56 mbox
-rw-r--r--    1 carlos  becas     40007 Sep 30 01:11 mi1
-rw-r--r--    1 root    other     40007 Sep 28 13:44 mied
drwxr-xr-x    3 carlos  becas       512 Sep 30 01:17 respaldo
dante% █
```

chmod (Permisos)

Éste comando nos permite modificar los permisos a los archivos y directorios, la asignación de permisos puede hacerse de dos formas, mediante el método simbólico o mediante el método octal.

Asignación de permisos:

chmod indica al sistema el cambio de permisos.

u Indica que se modificarán los permisos del usuario dueño del elemento.

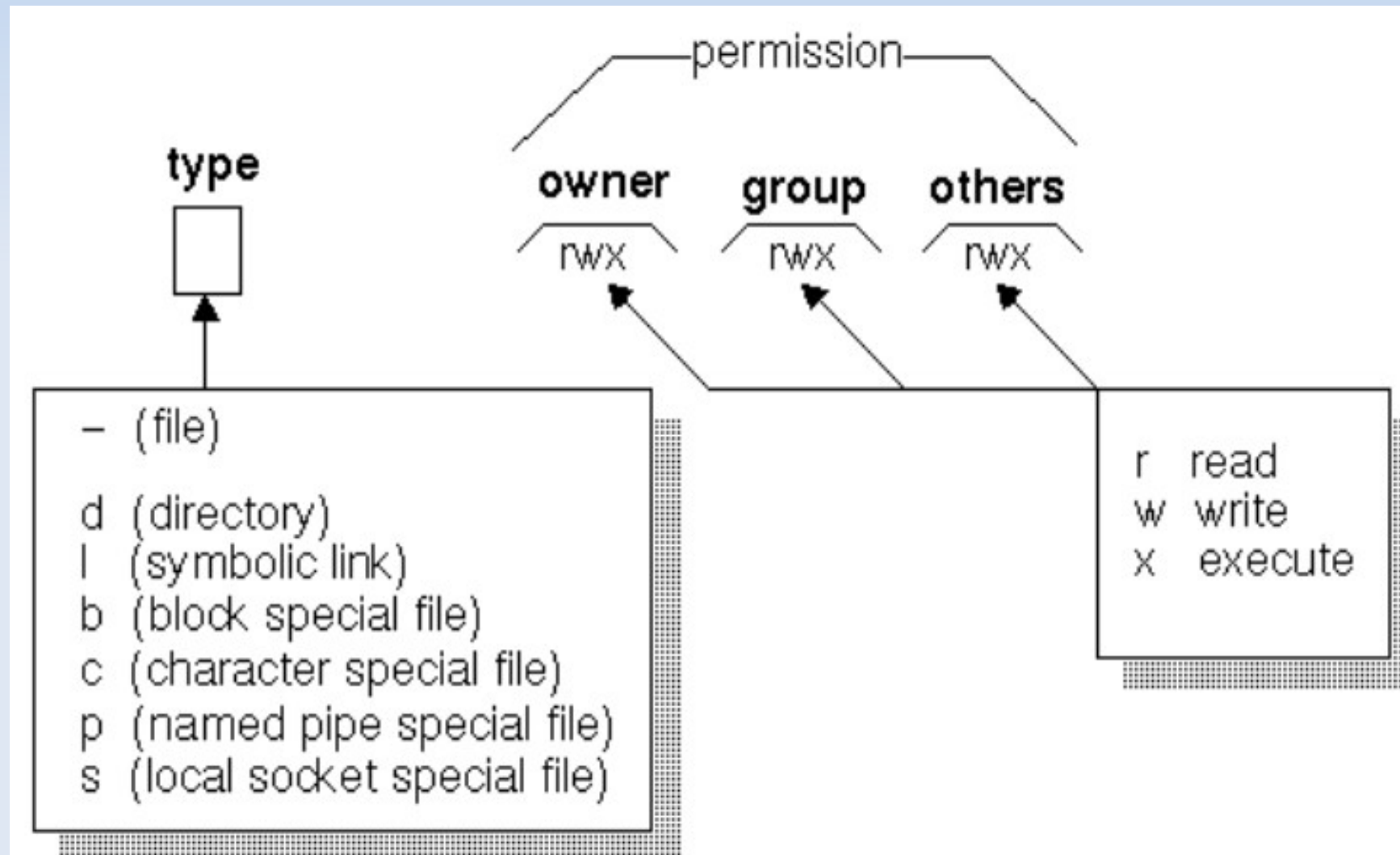
g Indica que se modificarán los permisos asignados al grupo.

o Indica que se modificaran los permisos de los otros usuarios que no pertenecen al grupo.

a Indica que afectará los permisos para todos los usuarios.

Permisos

Modo alfabético




```
chmod u+r,g-rx,o+r hola
```

Para modificar los permisos se pone el indicador de los permisos a afectar ya sea para el usuario, el grupo o los otros, seguido del signo + si se quiere agregar el permiso o de - si se quiere quitar el permiso separando con una coma los bloques de permisos, al final se indica el nombre del elemento a afectar.

Método octal

Con el método octal se asignan directamente los permisos en base a una tabla de validación que va numerada de 0 a 7 dependiendo de los permisos que se deseen asignar, cada permiso adquiere un valor numérico:

X vale 1

W vale 2

R vale 4

quedando nuestra tabla de la siguiente forma:

Asignación

Permisos

0	__ _	Ninguno
1	_ _ x	Ejecución
2	_ w _	Escritura
3	_ w x	Escritura y Ejecución
4	r _ _	Lectura
5	r _ x	Lectura y Ejecución
6	r w _	Lectura y Escritura
7	r w x	Lectura, escritura y ejecución

De tal forma que si deseamos poner todos los permisos para el usuario, permisos de lectura y ejecución para el grupo y ninguno para los otros el comando quedaría de la siguiente forma:

```
chmod 750 hola
```

Sustituyendo el número correspondiente para cada bloque de permisos al usuario, al grupo y a los otros.

Comandos para administrar los usuarios:

Adduser Agrega usuarios modo comando

Useradd Agrega usuarios modo comando

Userdel Elimina usuario

Usermod Modifica las propiedades del usuario

Groupadd Agrega grupos

Groupdel Elimina grupos

Groupmod Modifica grupos

Groups Lista los grupos a los que pertenece un usuario

Passwd Modifica propiedades de usuarios/grupos

who muestra un listado sencillo de quien se encuentra conectado en el servidor, a que hora se conecto y que esta haciendo.

w Muestra un listado con detalles de los usuarios conectados en el servidor.

whoami me indica con que login estoy registrado dentro del servidor, este comando es útil ya que es posible que un usuario se convierta en otro diferente utilizando el comando **su**

Arranque del sistema operativo LINUX

Arranque de Linux

El administrador de sistemas debe conocer las diferentes formas para dar de alta o de baja el sistema, de ello dependerá en gran medida el rendimiento y confiabilidad del sistema incluso después de un reinicio, también puede ser muy útil para rescatar un sistema cuando no inicia apropiadamente. Primero es necesario comprender lo que sucede cuando encendemos el equipo y el proceso de arranque del sistema.

Arranque de Linux

Lo primero que realiza el proceso de arranque de Linux es cargar una copia del kernel que se encuentra en el directorio `/boot`, a continuación detecta todos los dispositivos de hardware y los pone a disposición del sistema, busca en los scripts de configuración para leer el nivel de inicio para el cual esta configurado por default, luego busca los scripts del runlevel, finalmente ejecuta el script `rc.local`.

Arranque de Linux

Bios checa el sistema

Lanzamiento del Boot Loader (MBR)

segundo Boot Loader

/boot (kernel + initrd)

carga del Kernel en memoria y los diferentes módulos

Ejecución de /sbin/init

/etc/rc.d/rc.sysinit

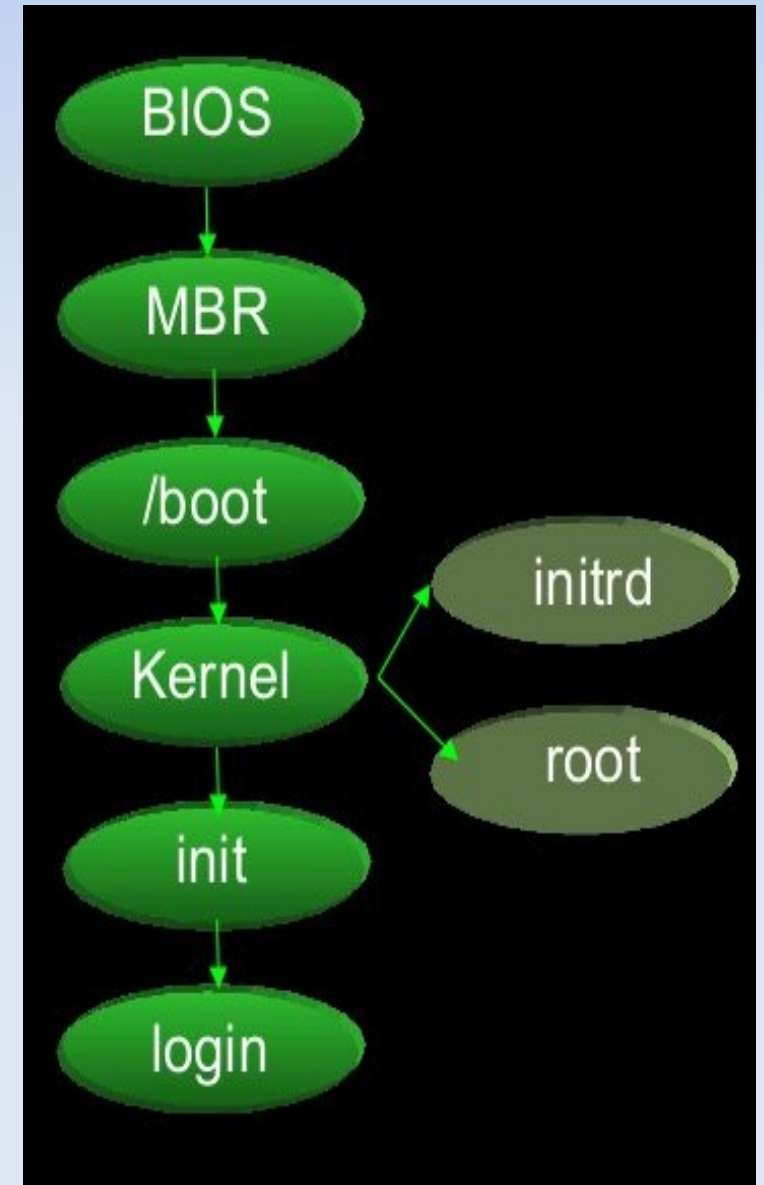
/etc/inittab

/etc/rc.d

/sbin/mingetty

Carga de servicios, aplicaciones y sistemas de ficheros

Secuencia de Login



Arranque de Linux

El nivel de inicio por default se encuentra definido en el archivo `/etc/inittab`, para ello necesitamos conocer los diferentes niveles que están predefinidos en el archivo de inicio:

- # These are the default runlevels

- # 0 = halt

- # 1 = single user mode

- # 2 = unused (but configured the same as runlevel 3)

- # 3 = multiuser mode (default Debian runlevel)

- # 4 = X11 with KDM/GDM/XDM (session managers)

- # 5 = unused (but configured the same as runlevel 3)

- # 6 = reboot

Arranque de Linux

Este programa de iniciación carga todos los servicios definidos, ejecuta los programas de iniciación y configuración y monta las particiones definidas.

/sbin/init lee la información del fichero de configuración /etc/inittab, carga en primer lugar los servicios básicos y luego los asociados al nivel de arranque elegido por el administrador.

- Se finaliza el arranque del sistema presentando al usuario el proceso de conexión (login) o el entorno gráfico (normalmente GNOME o KDE).

Arranque de Linux

Una vez que se ha cargado en memoria, el kernel ejecuta su primer programa init. Esta aplicación va a ser el padre de todos los procesos del sistema.

El comando `/sbin/init` utiliza la configuración que se encuentra en el archivo `/etc/inittab`, primero en búsqueda de `initdefault`, que le indica cual es el runlevel en el que debe entrar.

Un runlevel es una configuración que permite iniciar cierto grupo de programas

Cada línea contiene `id:runlevels:action:process`

Scripts de inicio

Cuando se inicia la maquina, se suelen ejecutar una serie de scripts básicos del sistema. En Debian GNU/Linux estos scripts son `/etc/init.d/rcS` y `/etc/init.d/rc`, este ultimo recibe como parametro el runlevel con el que inicia.

Estos scripts ejecutan otros scripts, que se encuentran en `/etc/rcS.d` y `/etc/rc[1-6].d`, respectivamente. Que a su vez son symlinks de scripts que estan en `/etc/init.d/`. Los symlinks tienen la forma `[SK][0-9][0-9]nombre`, la S o la K le dice si debe iniciar o detener ese proceso respectivamente. Esto lo hace ejecutando el script con el parametro `start` o `stop` segun corresponda. El número de 00-99 determina el orden de ejecucion.

Scripts de inicio

Servicios / Daemons

Entre los servicios que se inician al ejecutarse algun runlevel, existen algunos que podemos esperar que se encuentren en cualquier sistema UNIX.

Como el syslog, el cron, anacron, at y inetd

Apagar Linux

Para apagar el sistema podemos usar varios comandos:

`poweroff`

`halt`

`shutdown`

`reboot`

`init`

`telinit`

Aunque estos comandos nos permiten apagar o reiniciar el proceso puede variar, aquí hay que determinar cual es la forma más apropiada.

Apagar Linux

Shutdown

Este es considerado el comando más eficiente para detener, reiniciar el sistema, ya que su proceso de apagado es muy eficiente, de hecho “da de baja el sistema en una forma segura”. Se tiene control del apagado del sistema ya que se puede definir un tiempo, todos los usuarios son avisados de que el sistema se va a dar de baja mediante la instrucción SIGTERM, deshabilita el login para que nadie se pueda logear en el sistema, se comunica con el programa init y le solicita el nivel 0 de ejecución.

Apagar Linux

Shutdown

Este comando envía una señal de sincronización de discos, manda la señal de termino a todos los programas y los cierra eficientemente siguiendo su proceso normal de las aplicaciones, una vez que se han cerrado todos los procesos manda una nueva señal de sincronización para verificar que no hay tareas en ejecución, termina baja los demonios y finalmente apaga el sistema en forma segura.

Apagar Linux

<code>shutdown -h now</code>	Apaga el sistema en forma inmediata
<code>shutdown -r now</code>	Reinicia el sistema en forma inmediata
<code>shutdown -t sec acción</code>	Realiza la acción con t numero de segundos de espera .
<code>shutdown -k</code>	No apaga el sistema, solo envía un aviso a todos los usuarios conectados.
<code>shutdown -f acción</code>	Evita que se ejecute fsck al reinicio
<code>shutdown -F acción</code>	Fuerza a que se ejecute fsck al reinicio
<code>shutdown -c</code>	Cancela la instrucción shutdown
<code>Shutdown +m acción</code>	Espera m minutos antes de realizar la acción

Runlevel

Para saber en que nivel se esta ejecutando el sistema puede ejecutar el comando runlevel:

```
sesion1@milo:~$ /sbin/runlevel
```

N 2

Aquí nos dice que el sistema se inicio en modo 2.

Monousuario Linux

Init

Si deseamos cambiar el modo de ejecución del sistema podemos hacerlo directamente con el comando `init` y el numero de inicio, por ejemplo si deseamos cambiar a nivel de ejecución 1

```
init 1
```

El sistema ejecutara los scripts correspondientes para cerrar los demonios, los dispositivos de red y quedará en modo monousuario.

```
sesion1@milo:~$ /sbin/runlevel
```

```
3 1
```

Aquí el sistema nos dice que inició en modo 3 pero se reinicio en modo 1, ahora podemos comenzar nuestras tareas administrativas.

Introducción al Sistema Operativo LINUX

Reconocimiento del servidor y del sistema hostname

hostname nos muestra el nombre del servidor.

- i Esta opción nos muestra la dirección ip que tiene asignada el servidor de trabajo.

- d Nos muestra el dominio del servidor.
- f Muestra un listado completo incluyendo el nombre del servidor y su dominio.

uname

uname Nos indica el tipo de LINUX que se esta utilizando.

linux

- n Indica el nombre del servidor
- r Muestra la versión del kernel
- m Indica la arquitectura del procesador
- a Muestra todos los datos antes mencionados

pwd

(Print Working Directory) éste comando imprime en pantalla el directorio actual de trabajo, es decir, donde nos encontramos posicionados dentro del sistema de directorios indicándonos la ruta absoluta.

touch

Este comando nos permite crear archivos vacíos, de tamaño 0.

touch prueba

Este comando es útil cuando nos interesa generar archivos que tienen una función especial como las bitácoras.

Caracteres Especiales.

UNIX maneja varios caracteres especiales, cada uno de ellos realiza una función en específico, en ocasiones pueden ser combinados para aumentar el potencial del comando en ejecución o para redireccionamiento, se recomienda que estos caracteres no se utilicen al nombrar archivos o directorios.

Estos caracteres son:

`; , < > | >> ^ $ & / ~ \ [] ' " () . * ?`

Comodines *, ?

Existen dos caracteres llamados comodines los cuales pueden sustituir un carácter o conjunto de caracteres durante la ejecución de comandos, Ej.

`ls ??a` Muestra aquellos archivos y directorios los cuales tengan el nombre de tres letras y la tercera sea la letra a

`ls *u*t*` Muestra aquellos archivos y directorios los cuales tengan en el nombre una u y una t en cualquier posición pero en el orden que se le indica.

Rutas Absolutas Y Relativas

Las rutas absolutas siempre inician con el símbolo de / que indica el punto más alto de la estructura de directorios, a partir de aquí se debe de poner la ruta completa para llegar al punto que se quiera afectar.

```
cd /home
```

```
cd /var/spool/mail
```

Las rutas relativas nunca inician con el símbolo de raíz / en lugar de ello inician la ruta a partir del directorio actual de trabajo.

Supongamos que se encuentra en su directorio home y quiere regresar dos directorios hacia arriba para posicionarnos en la raíz del sistema:

```
cd ../..
```

rm (Remove)

El comando rm borra archivos, puede ser borrado un solo archivo o varios haciendo uso de los metacaracteres * ?

rm saludo

rm lo*

rm pr??ba

rm -r directorio

La opción -r hace un borrado en forma recursiva, esta opción es de mucho cuidado ya que se borrará el directorio especificado y todo su contenido.

rm -f *

La opción -f fuerza a que los elementos sean borrados aunque nosotros no tengamos permisos sobre ellos.

mv (move)

El comando mv tiene dos funciones básicas, la primera es mover archivos o estructuras de directorios de un lugar a otro, la segunda es renombrar archivos o directorios.

```
mv saludo carta/
```

Aquí se está moviendo el archivo saludo dentro del directorio carta.

```
mv saludo hola
```

Aquí se está cambiando el nombre del archivo saludo, ahora se llamará hola.

Trabajo con archivos

cat nos permite ver el contenido de uno o mas archivos, su función original es concatenar archivos.

```
cat archivo1 archivo2 ...
```

-n Utilizando la opción el comando imprime el contenido del o los archivos numerando las líneas en forma ascendente.

```
cat -n archivo1 archivo2 ...
```

head

nos muestra las primeras 10 líneas contenidas en un archivo

```
head nom_archivo
```

-n esta opción le indica al comando que despliegue las primeras n número de líneas en lugar de 10.

```
head -20 nom_archivo
```

tail

nos muestra las últimas 10 líneas contenidas en un archivo

```
tail nom_archivo
```

-n Esta opción le indica al comando que despliegue las últimas n número de líneas en lugar de 10.

```
tail -20 nom_archivo
```

pico

Este es un programa que permite editar archivos de texto, es muy fácil de utilizar, ya que presenta un menú de comandos en la parte inferior de la pantalla y no requiere de muchos comandos para poder editar el contenido del archivo.

Permite el desplazamiento del cursor mediante las teclas de navegación.

ctrl+o permite guardar el archivo, si ya tiene nombre el archivo lo guarda automáticamente, si el archivo no tiene nombre nos va a pedir el nombre con el cual lo deseamos guardar.

Filtros

grep

Este comando busca patrones en archivos. Por defecto devuelve todas las líneas que contienen un patrón determinado en uno o varios archivos. Utilizando las opciones se puede variar mucho este comportamiento. Si no se le pasa ningún archivo como argumento hace la búsqueda en la entrada estándar.

Sintaxis:

grep [opciones] <patrón> [ficheros]

Algunas opciones:

-c : devuelve sólo la cantidad de líneas que contienen al patrón.

-i : ignora las diferencias entre mayúsculas y minúsculas.

-H : imprime además de las líneas, el nombre del archivo donde se encontró el patrón. Es así por defecto cuando se hace la búsqueda en más de un archivo.

-l : cuando son múltiples archivos sólo muestra los nombres de aquellos donde se encontró al patrón y no las líneas correspondientes.

-v : devuelve las líneas que no contienen el patrón.

-r : busca en un directorio de forma recursiva.

-n : imprime el número de cada línea que contiene al patrón.

Ejemplos:

```
$ grep -H -r linux /usr/share/doc
```

```
# grep -n error /var/log/messages
```

```
$ grep -i ftp /etc/passwd
```

```
$ grep -c root /etc/group
```

```
$ grep -l -r -i img /var/www/icons/*
```

Algunos programas útiles dentro de LINUX

fortune Elige de una base de datos el mensaje del día como el que aparece en pantalla cada vez que iniciamos sesión.

date muestra la hora del sistema.

df Muestra el estado de las particiones de los discos duros indicándonos el espacio utilizado y el estado libre en disco.

du muestra el espacio utilizado entre archivos y directorios a partir del directorio desde donde se ejecutó el comando.

last muestra las últimas conexiones de los usuarios al servidor.

last -20 muestra los últimos 20 registros al servidor.

last -20 usuario muestra los últimos 20 registros al servidor del usuario carlos

sort Ordena el contenido de un archivo por líneas en orden alfabético.

`sort -r` ordena el contenido de un archivo en forma inversa alfabéticamente.

`wc` cuenta las líneas, palabras y caracteres contenidos en un archivo.

`wc -l` Cuenta únicamente las líneas

`wc -w` Cuenta únicamente las palabras

`wc -c` Cuenta únicamente los caracteres

Respaldos

tar

tar nos permite empacar una estructura de directorios, su función principal es almacenar el contenido de un directorio en forma recursiva dentro de un archivo.

Sintaxis:

```
tar [opciones] archivo.tar archivo
```

Principales opciones de tar

- c crea un nuevo archivo tarareado.
- v verbose, imprime en pantalla todo aquello que esta afectando.
- f Indica al sistema que va a afectar a un archivo
- x Extrae el contenido de un archivo .tar

tar -cvf archivo.tar archivo

Aquí se está creando un archivo utilizando la opción -c la opción -v muestra en pantalla todos los archivos y directorios que está afectando, la opción -f indica que se va a enviar la salida del comando a un archivo, a continuación de las opciones se especifica el nombre que va a tener el archivo tarareado y por último el nombre del directorio que va a ser afectado.

`tar -xvf archivo.tar`

Aquí se está recuperando la información almacenada dentro de un archivo tarreado, con la opción `-x`, una de las principales características de `tar` es que no le importa quien es el dueño del archivo, el grupo al que pertenece ni los permisos que tuvo el archivo al momento de ser tarreado, ya que cuando realizamos el proceso de destarrear toda la información afectada toma los permisos que nosotros tenemos por default.

Como regla general cualquier opción es combinada con las opciones -vf ya que es recomendable ver qué está realizando el sistema, así como para indicarle que se está enviando la salida del comando a un archivo, de lo contrario tomará por default la unidad de cinta.

-t Lista todos los archivos contenidos en un archivo .tar en formato largo mostrando todos sus detalles.

```
tar -tvf archivo.tar
```

-u update, actualiza el contenido del archivo tar realizando una búsqueda dentro de la estructura de directorios del sistema desde donde se inició el proceso de tarareo, añadiendo todos los archivos y directorios que hayan sido creados o sufrido modificaciones después de haber sido creado el archivo tarareado.

tar -uvf archivo.tar

-w Esta opción nos pide confirmación para tararear o destararear cada elemento que va a ser afectado.

```
tar -cwvf archivo.tar
```

```
tar -xwvf archivo.tar
```

-d Opción no disponible en todas las versiones, hace una búsqueda de todos los archivos y directorios que hayan sufrido modificaciones después de haber sido creado el archivo .tar

```
tar -dvf archivo.tar
```

gzip Permite realizar una compresión de archivos, manejando 9 niveles de compresión que van desde el -1 al -9 donde el nivel 1 es el nivel de compresión más bajo mientras el nivel -9 es el nivel de compresión más alto.

Al momento de ejecutar esta utilera el programa le añade la extensión .gz al archivo comprimido, por default maneja el nivel de compresión 6.

Sintaxis: `gzip [-n] archivo`

La opción `-n` indica el nivel de compresión que se quiera realizar, el nivel óptimo es el `-9` ya que realiza la compresión más alta que puede tener el archivo afectado.

`-d` Esta opción permite descomprimir un archivo gzipeado.

Sintaxis: `gzip -d archivo`

-f Fuerza la compresión o descompresión de la información afectada sin importar los permisos o propiedades que tenga el archivo o directorio al momento de ser afectado.

Sintaxis: `gzip [-fn] archivo`

-t Ésta opción hace una revisión del archivo tarareado para verificar su integridad, en caso de que el archivo no se encuentre íntegro el sistema mandara un mensaje indicando el tipo de incongruencia del archivo.

Sintaxis: `gzip -t archivo.gz`

-l Ésta opción muestra las propiedades del archivo tarareado indicando el tamaño del archivo cuando esta comprimido, el tamaño de la información cuando se encuentra descomprimida y el porcentaje de compresión de la información.

```
gzip -lv archivo.tar.gz
```

Ligas

Existen dos tipos de ligas, una de ellas es la de tipo suave y la otra es de tipo dura.

La liga de tipo suave es un vínculo hacia un archivo pero la liga es de tamaño muy pequeño ya que solo hace referencia al archivo que se esta ligando, funciona como un acceso directo y su función generalmente es asociar a un programa de nombre complejo con una liga la cual tiene un nombre más fácil de recordar.

Sintaxis de la liga suave

`In -s nom_archivo nom_liga`

El comando `In` nos permite generar las ligas, en este caso con la opción `-s` se indica al sistema que se esta generando una liga de tipo suave.

Liga dura

La liga dura al igual que la liga suave mantiene una relación con un archivo el cual esta ligado, la diferencia es que al comparar el tamaño de la liga con el archivo, ambos tienen el mismo tamaño, al editar el contenido de uno se actualiza automáticamente el otro, pero a comparación de un archivo común es que la liga esta asociada al mismo inodo del archivo con el que esta ligado.

Sintaxis de la liga dura

`In nom_archivo nom_liga`

En este caso se esta generando una liga dura, al aplicar el comando `In` sin opciones el sistema interpreta el comando como la creación de una liga dura.