

**Facultad de Ingeniería de Sistemas, Cómputo y
Telecomunicaciones**

**PARTE PRÁCTICA
Sistemas Operativos**

Santiago Raúl Gonzales Sánchez

UNIDAD I

CONCEPTOS Y COMANDOS BÁSICOS

La unidad tiene como propósito que el estudiante comprenda los conceptos y fundamentos necesarios sobre el sistema GNU/Linux, valorando la importancia en el área de computación y sistemas. Contiene:

- Fundamentos de GNU/Linux
- Trabajando con GNU/Linux
- Comandos Básicos
- Editor de Texto VIM

Lección 1

Fundamentos de GNU/Linux

1.1. ¿Qué es GNU/Linux?

GNU/Linux es una versión de Unix de distribución libre desarrollado originalmente en 1991 por Linus Torvalds, en ese tiempo estudiante de la universidad de Helsinki [3].

GNU/Linux inició la distribución libre del sistema operativo de Internet, dando comienzo a uno de los fenómenos más evolucionarios en el mundo de la informática y computación [3].

GNU/Linux se inspira en la versión Unix también libre, el sistema operativo Minix de Andrew Tanenbaum [3].

1.2. Características de GNU/Linux

De acuerdo a [3] [4] se describe las siguientes características:

- Es un sistema operativo multitarea y multiusuario. Varios usuarios pueden conectarse a un mismo ordenador a la vez y ejecutar distintos programas al mismo tiempo.
- Compatible con casi todos los sistemas Unix existentes a nivel de código. Incluyendo los IEEE POSIX.1, System V y BSD. La capacidad de transportar los programas fue una de las reglas de diseño del sistema. De manera que gran parte de las posibilidades de Linux las encontramos en el resto de los Unix.
- Puede coexistir en entornos que disponen de otros sistemas operativos instalados.
- En GNU/Linux como en la mayoría de los sistemas UNIX, después de una instalación original, puede instalar o remover software sin la necesidad de tener que reiniciar el equipo.
- Puede iniciar y detener servicios individuales (como servidores Web, FTP y servicios de correo) sin reiniciar o interrumpir el trabajo que otros usuarios realizan en el sistema.
- Si las aplicaciones que quiere no están desarrolladas con la versión de su sistema Linux, puede descargar e instalarlos con un simple comando usando herramientas como apt, yum y rpm.
- Protección de la memoria entre procesos, de manera que uno de ellos no pueda colgar el sistema.
- La memoria es gestionada como un recurso unificado para los programas de usuario y para el caché de disco, de tal forma que toda la memoria libre puede ser usada para caché y ésta puede a su vez ser reducida cuando se ejecuten grandes programas.
- Capacidad de trabajo en red, donde diversos protocolos de red están incluidos en el Kernel: TCP, IPv4, IPv6, AX.25, X.25, IPX, DDP, Netrom, etc.

1.3. Dispositivos y periféricos en GNU/Linux

Para GNU/Linux todo es un archivo, incluyendo dispositivos como discos duros, cdroms, disquetes, memorias usb, etc., así como dispositivos de comunicación como puertos seriales y paralelos, módems, etc, incluso también las consolas o terminales son dispositivos asociados a un archivo. Estos dispositivos son enlazados a un dispositivo de archivo, es decir un dispositivo físico es representado o asociado a un archivo. Estos archivos se encuentran dentro del directorio /dev [2].

Los dispositivos en GNU/Linux son identificados con dos o tres letras, además si el dispositivo admite particiones se utiliza una progresión numérica o alfabética para identificar la partición [2]. En la siguiente tabla se indica el tipo dispositivo y su descripción.

Tipo	Dispositivo
hd	Discos duros IDE
sd	Discos duros SCSI
scd	Cdrom SCSI
fd	Unidades de disquetes
lp	Puertos paralelos
tty	Terminales o consolas
pts	Terminales remotas o de red, incluyendo las abiertas en Window X
ttyS	Puertos seriales
eth	Tarjetas o interfaces de red ethernet

Los dispositivos que admiten particiones generalmente éstas se designan con letras, así por ejemplo las unidades IDE, que son las más comunes en cualquier PC actual. Su designación sería la siguiente:

Disco Duro	Dispositivo Linux
Primario Maestro	/dev/hda
Primario Esclavo	/dev/hdb
Secundario Maestro	/dev/hdc
Secundario Esclavo	/dev/hdd

En GNU/Linux el dispositivo /dev/hda representa al disco duro, particionado se numera secuencialmente a partir de 1 en cada partición de la siguiente manera:

Partición en GNU/Linux	Equivalente en Windows
/dev/hda1	C:
/dev/hda2	D:
/dev/hda3	E:
/dev/hda4	F:

Los dispositivos que no admiten particiones en sus sistemas de archivos, tales como disquetes o cdrom, se numeran secuencialmente a partir de 0 o simplemente se omite. Ejemplos:

- Disquete equivalente a: en Windows, en Linux: /dev/fd0
- Unidad de cdrom en secundario maestro: /dev/cdrom

En cuanto a puertos seriales, sus equivalentes con Windows serían los siguientes:

- Windows COM1, Linux /dev/ttys0
- Windows COM2, Linux /dev/ttys1

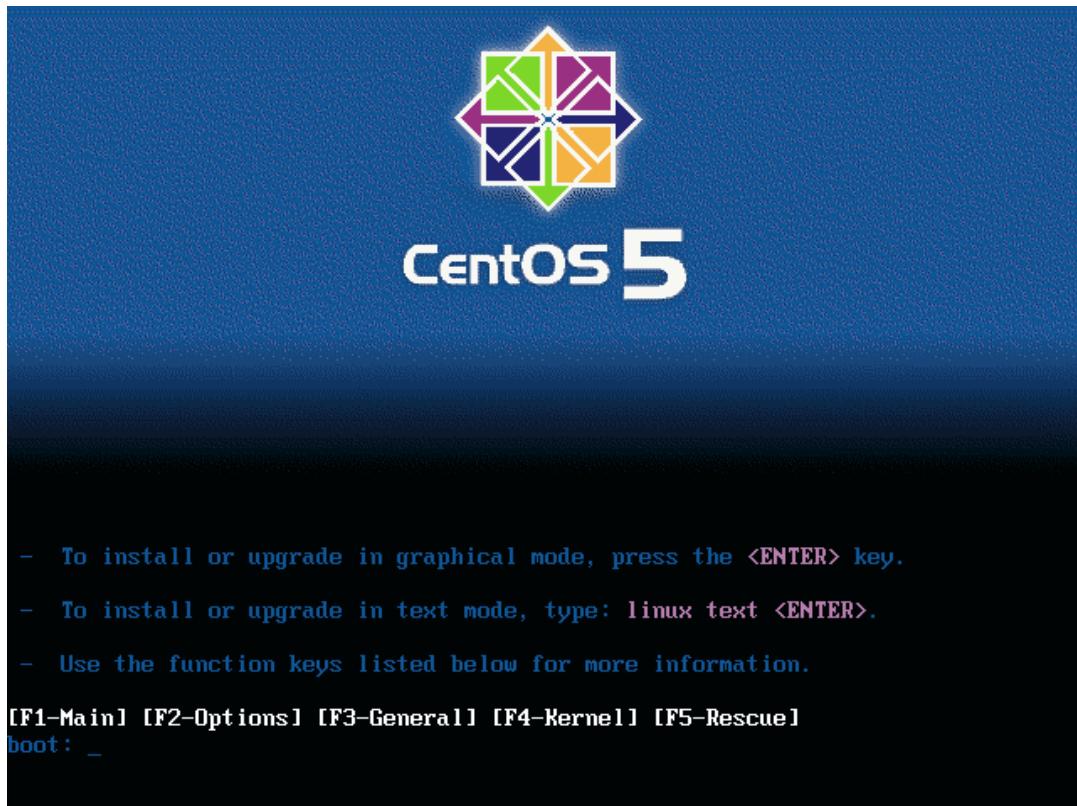
Puertos paralelos se designan de la siguiente manera:

- Primer puerto paralelo, Windows LPT1, Linux /dev/lp0
- Segundo puerto paralelo, Windows LPT2, Linux /dev/lp1

Los dispositivos se numeran con su identificador de dispositivo y secuencialmente a partir de 0 y hasta donde la arquitectura de hardware lo limite.

1.4. Instalando un GNU/Linux. CENTOS 5.4.

1. Inserte el primer disco de instalación de CentOS 5.4 y cuando aparezca la interfaz de diálogo de inicio (boot) presione la tecla **ENTER**.



2. Si desea verificar la integridad de los discos a partir del cual se realizará la instalación, seleccione **<OK>** y pulse la tecla **ENTER**. Si está seguro de que el disco o disco de instalación se encuentran en buen estado, seleccione **<Skip>** y pulse la tecla **ENTER**.



3. Pulse el botón **Next** para continuar la instalación de **CentOS**.

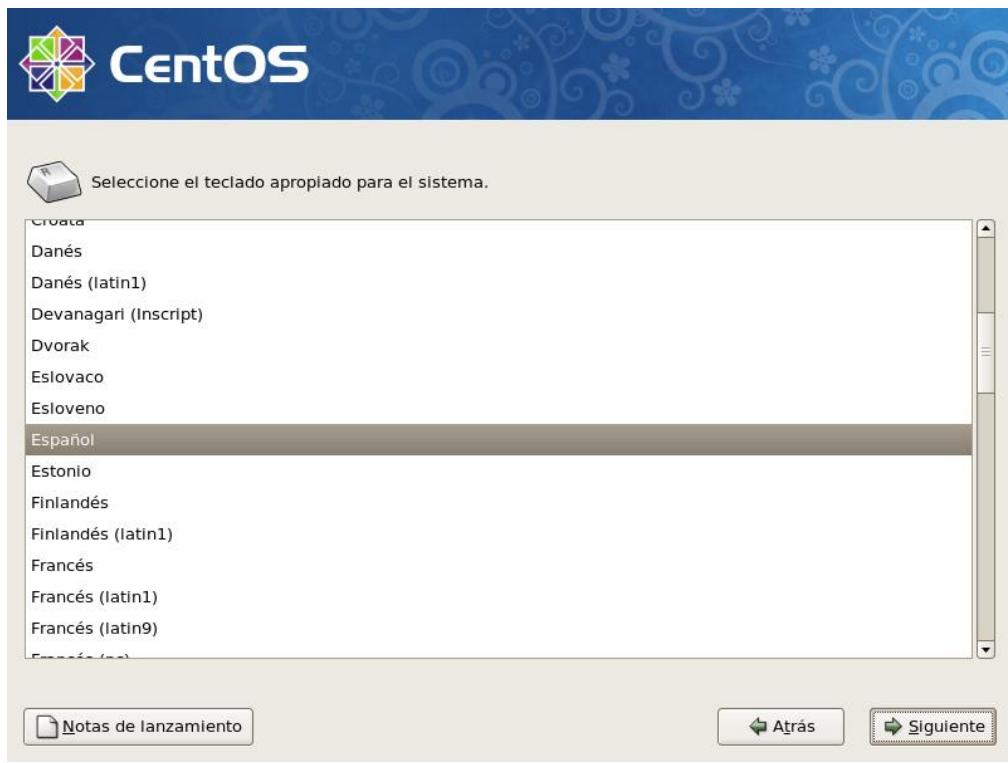


4. Seleccione el tipo de idioma que utilizará durante el proceso de instalación (en nuestro caso se seleccionará Spanish) y luego presione el botón **Next**.





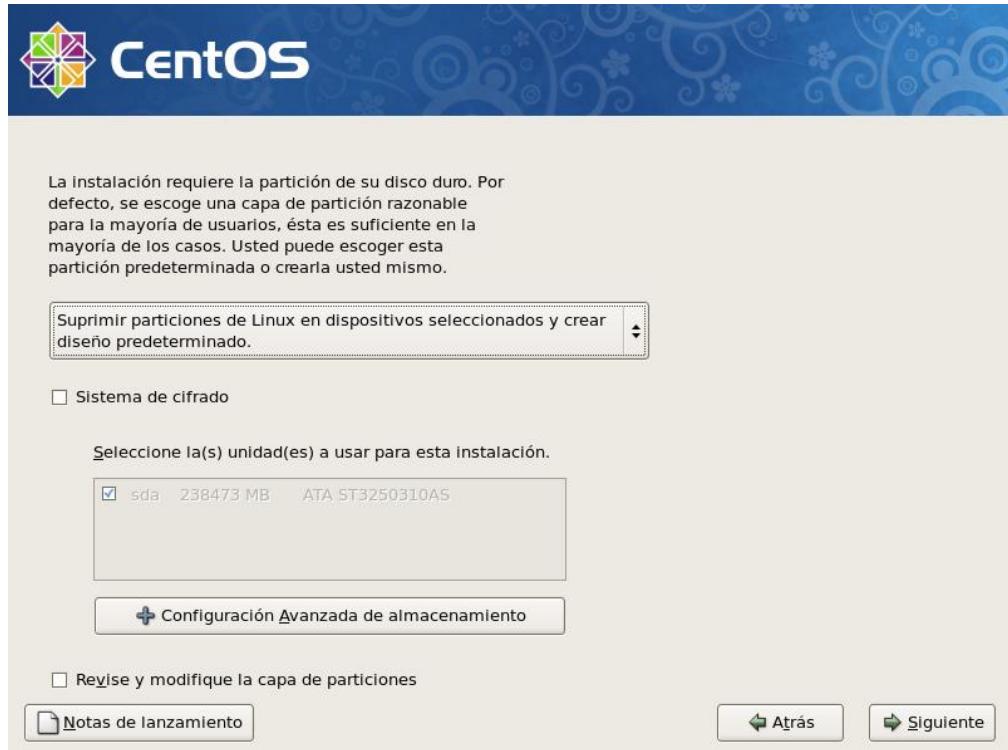
5. Seleccione el teclado apropiado para el sistema y presione el botón **Siguiente**.



6. Para crear las particiones de forma automática puede seleccionar:

- Suprimir particiones en dispositivos seleccionados y crear disposición.
- Suprimir particiones de linux en dispositivos seleccionados y crear diseño predeterminado.
- Usar espacio disponible en dispositivos seleccionados y crear disposición.

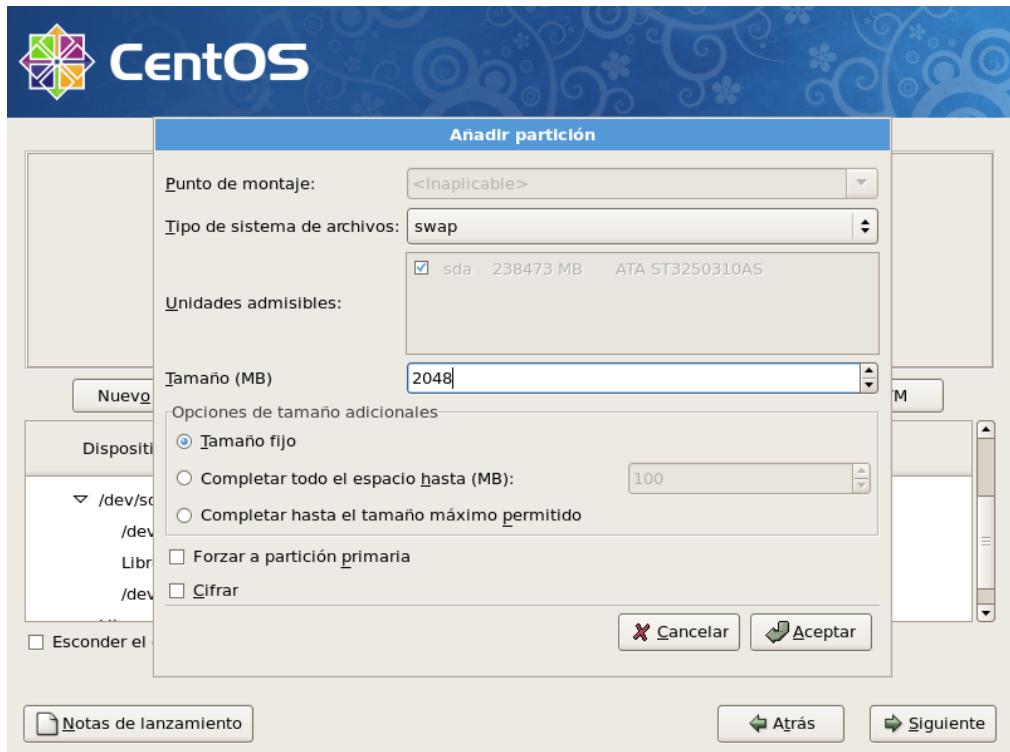
Para un mejor control, es recomendable que uno mismo cree las particiones, por ello seleccionaremos la opción <**Crear diseño personalizado**>. Presione el botón **Siguiente**.



- Realizado el paso anterior ingresará a la herramienta para gestionar las particiones del disco duro. En caso de compartir con otro sistema operativo, deberá disponer de un **Espacio Libre** (sin formatear).



8. Para crear la nueva partición presione el botón **Nuevo**. Crear la partición para la memoria virtual (**swap**), el punto de montaje <**Inaplicable**>, tipo de sistema de archivo <**swap**>, el tamaño recomendable al doble de la memoria RAM. Luego presione el botón **Aceptar**.



9. Realizado el paso anterior presione el botón **Nuevo**. Crear la partición para la raíz (/), el punto de montaje deberá indicar </>, tipo de sistema de archivo <**ext3**>,

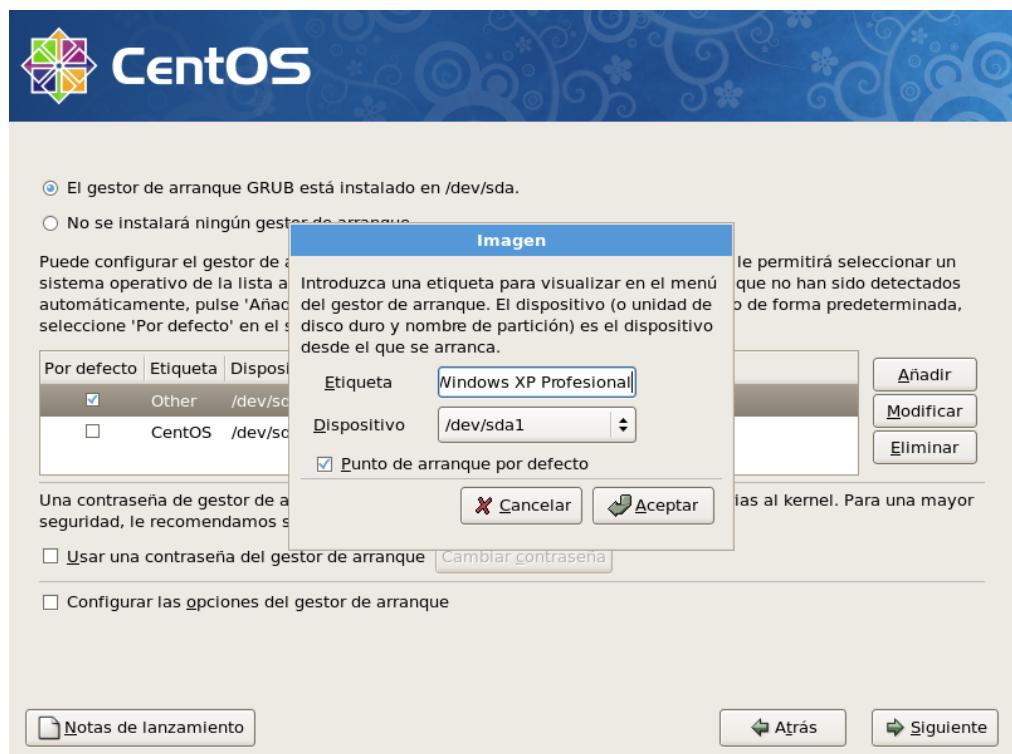
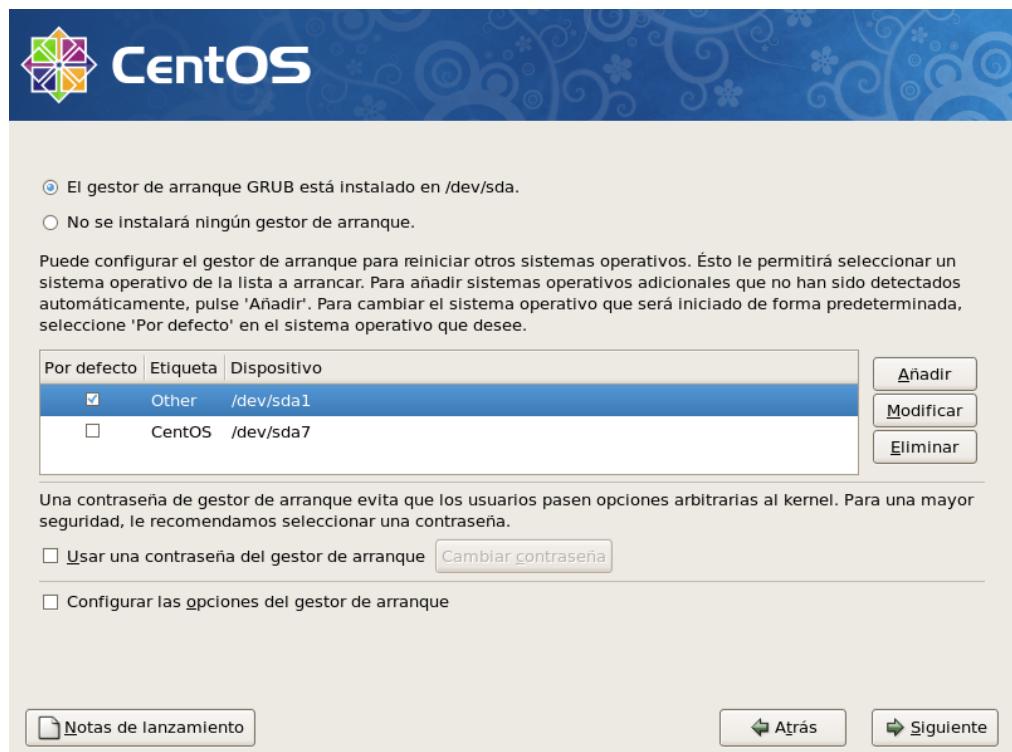
el tamaño a seleccionar es <**Completar hasta el tamaño máximo permitido**>. Luego presione el botón **Aceptar**.



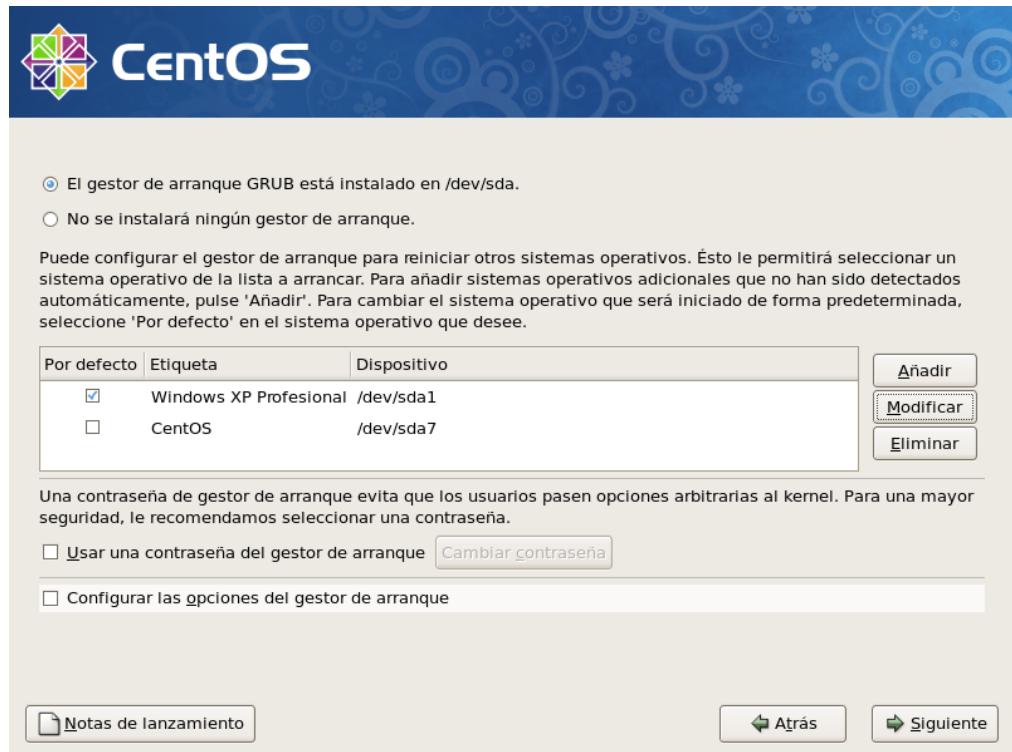
10. Al concluir le mostrará la tabla de particiones actualizada. Si está conforme, presiona el botón **Aceptar y**, luego, **Siguiente** para pasar a la siguiente ventana.



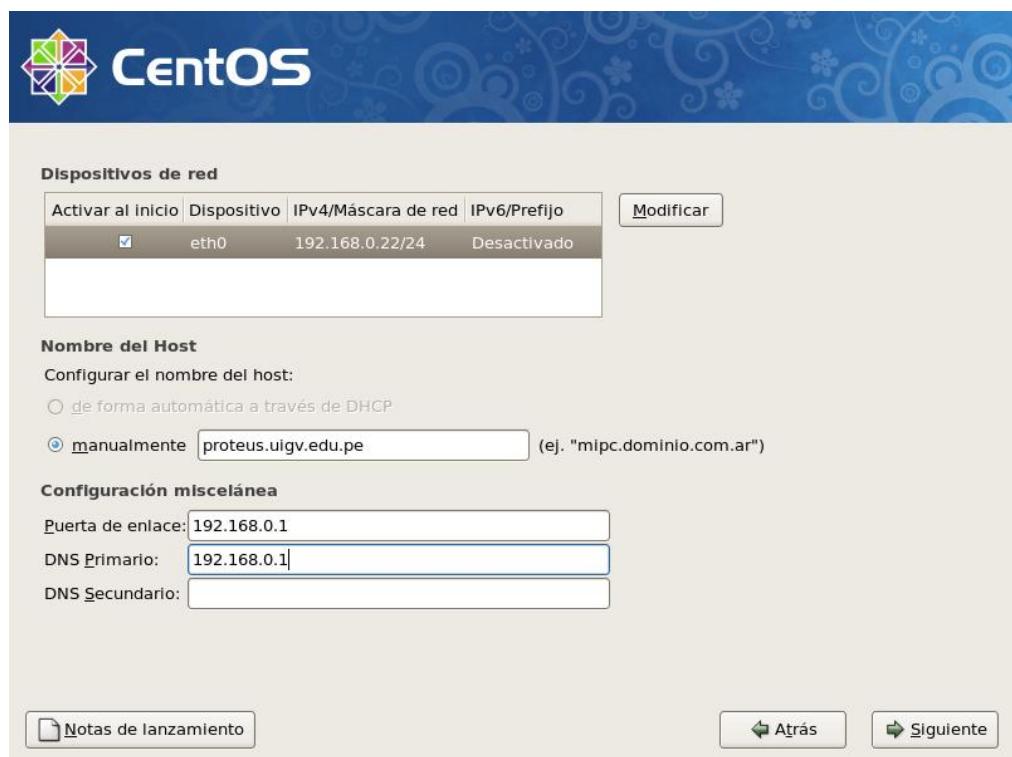
11. Configurar el gestor de arranque **GRUB** para iniciar con otros sistemas operativos. Para definir las etiquetas, presionar el botón **Modificar**.



12. El gestor de arranque **GRUB** quedará de la siguiente manera. Si es correcto, presione el botón **Siguiente**.



13. Para configurar la interfaz de red, deberá presionar el botón **Modificar** y completar los datos. Luego ingrese los datos del Nombre del Host, Puerto de Enlace y DNS Primario. Si los datos son correctos, presione el botón **Siguiente**.



14. Seleccionar la región.



15. Ingrese una **clave** de acceso para el usuario **root** (administrador del sistema). Deberá escribirla dos veces para confirmarla. Se recomienda ingresar una clave utilizando alfanumérico. Presione el botón **Siguiente**.



16. Defina el grupo de paquetes para la instalación del sistema. Para un mejor control, seleccione **Personalizar ahora**. Presione luego el botón **Siguiente**.



17. En la siguiente ventana, seleccione los paquetes (Lado izquierdo grupo de paquetes y en el lado derecho los paquetes que contiene este grupo). Presione el botón **Siguiente**.



18. Para iniciar la instalación, presione el botón **Siguiente**.



19. Si para la instalación utiliza varios discos, le solicitará los CD por orden de numeración. A continuación, presione el botón **Continuar**.



20. Realizado el paso anterior, se iniciará el proceso de instalación.



21. Finalizado la instalación, deberá presionar el botón **Reiniciar**.



Lección 2

Trabajando con GNU/Linux

2.1. Iniciando Sesión

El siguiente paso es ingresar al sistema operativo GNU/Linux. Para ello deberá escribir el nombre de usuario y la contraseña. Ingrese la cuenta de administrador desde la pantalla gráfica de conexión, teclee **root** en el intérprete de comandos y luego presione la tecla **[Enter]**, escriba la contraseña de root que ingresó durante la instalación y presione la tecla **[Enter]**.

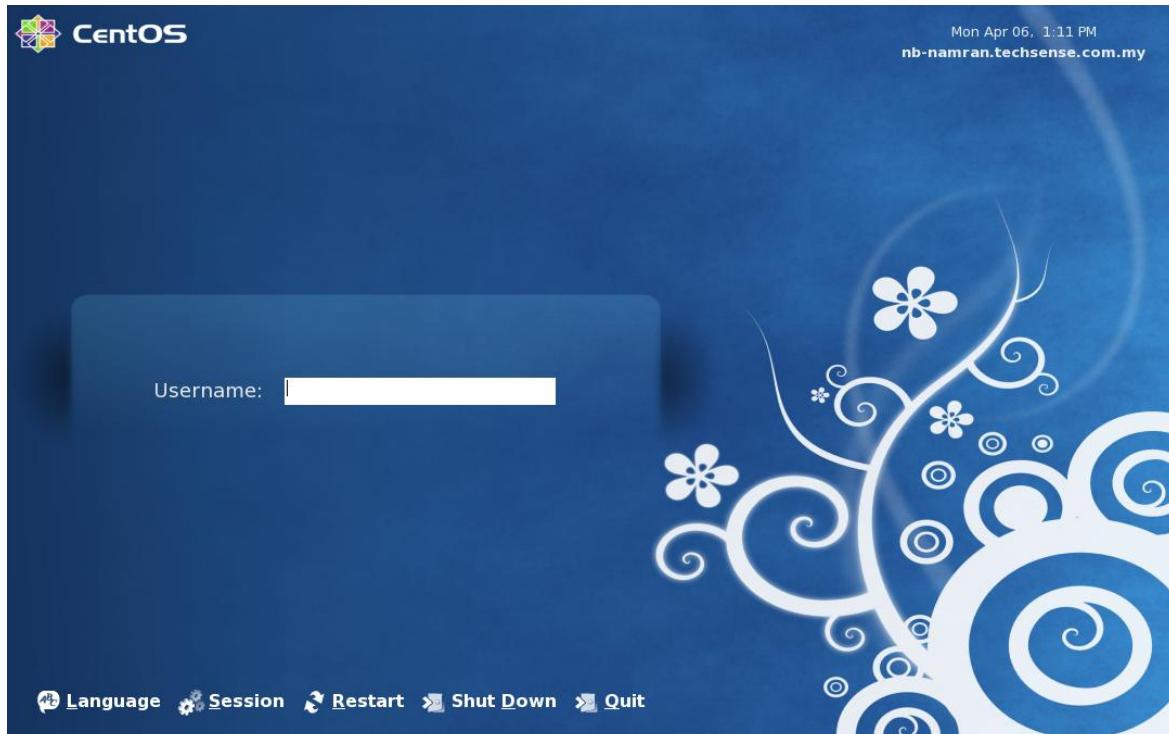


Figura 2.1. Pantalla de Ingreso al Sistema CentOS

Para conectarse como un usuario normal (en caso de haberlo creado), escriba su nombre de usuario en el indicador de comandos (login prompt) de conexión y presione **[Enter]**, escriba la contraseña que seleccionó cuando creó la cuenta de usuario y presione **[Enter]**.

A diferencia de otros sistemas operativos su sistema GNU/Linux utiliza cuentas para administrar privilegios y mantener la seguridad. No todas las cuentas son creadas de la misma manera, algunas tienen menos privilegios para acceder a los ficheros o ejecutar algún tipo de servicio.

En GNU/Linux el uso de las mayúsculas y minúsculas son distintas, lo que significa que escribiendo root se refiere a una cuenta diferente que ROOT. Por defecto el usuario root se refiere al administrador del sistema o superusuario.

Nota.- CentOS crea una cuenta (**root**) durante la instalación, los usuarios nuevos podrían estar tentados a usar sólo esta cuenta para todas sus actividades. Esto no es una buena idea dado que la cuenta root puede hacer cualquier cosa en el sistema, pudiendo dañar fácilmente su sistema, borrando por error o modificando ficheros que pertenecen al sistema. [1]

Una vez que ingrese al entorno gráfico o sistema X Window, encontrará una interfaz gráfica conocida como un *escritorio*.

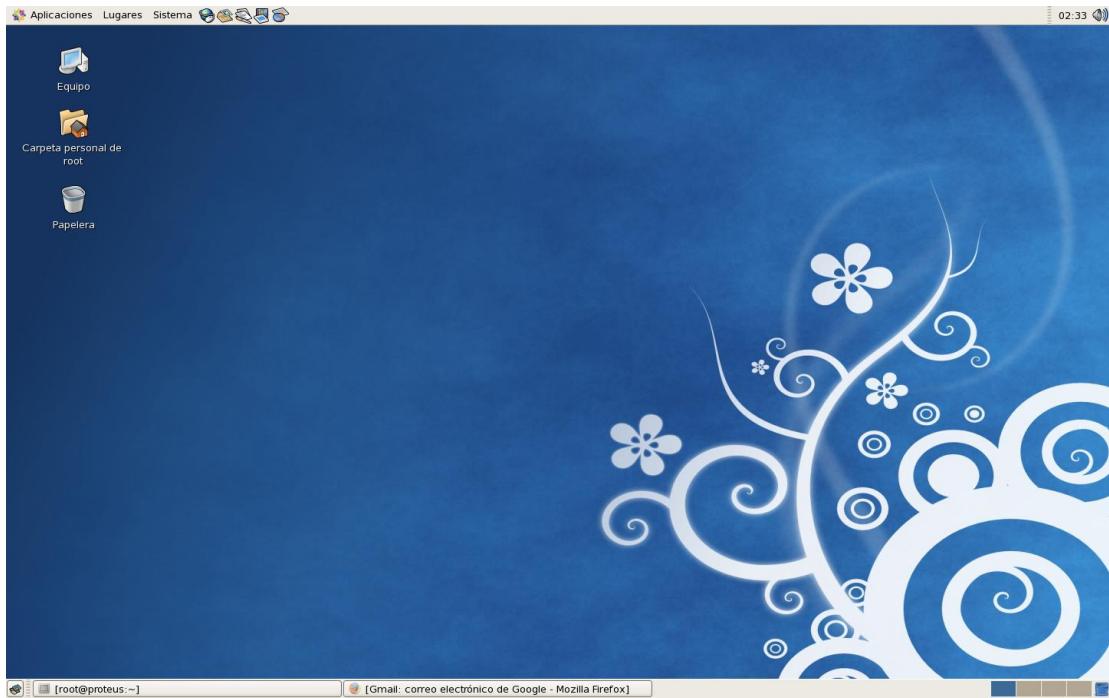


Figura 2.2. El escritorio gráfico de CentOS

2.1.1. Abrir una ventana de terminal

El escritorio le ofrece acceso a un *intérprete de comandos*, una aplicación que le permite escribir comandos en vez de utilizar la interfaz gráfica para todas las actividades a realizar.

Puede abrir un intérprete de comandos dando un clic con el botón derecho del mouse sobre el escritorio y luego del menú seleccionar abrir terminal.

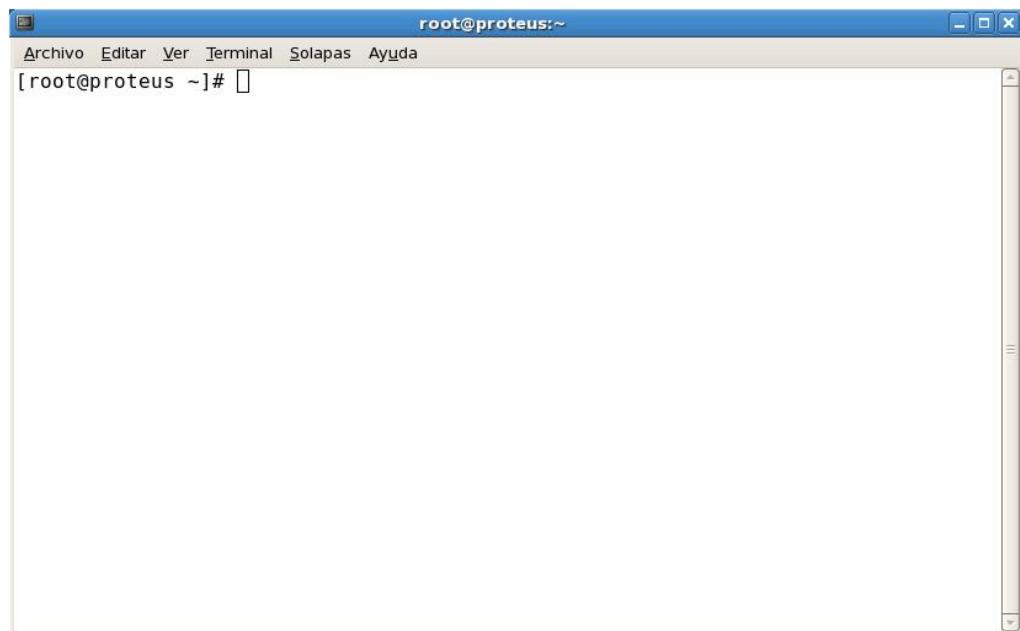


Figura 2.3. Terminal en CentOS

En el mismo terminal puede abrir otra ventana de intérprete de comando. Para ello deberá ir al menú Archivo y luego clic en nueva solapa.

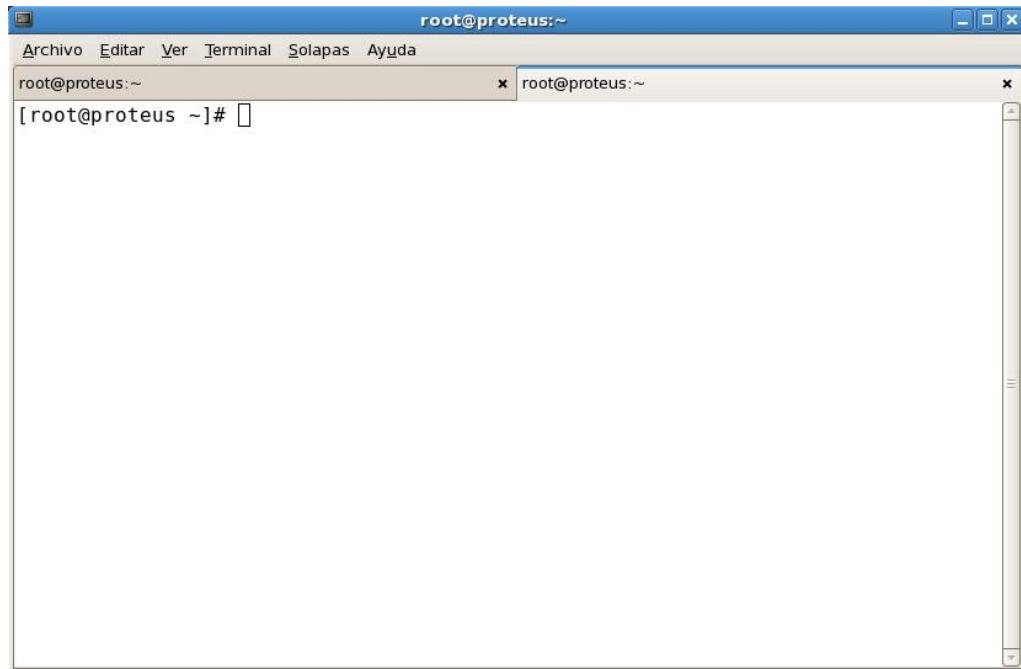


Figura 2.4. Terminal en CentOS con dos sesiones

Una de las características principales de GNU/Linux es poder abrir tantas sesiones utilizando la misma cuenta de usuario.

2.1.2. Cerrar terminal o sesión

Para salir del intérprete de comandos escriba el comando **exit** en el indicador de comandos.

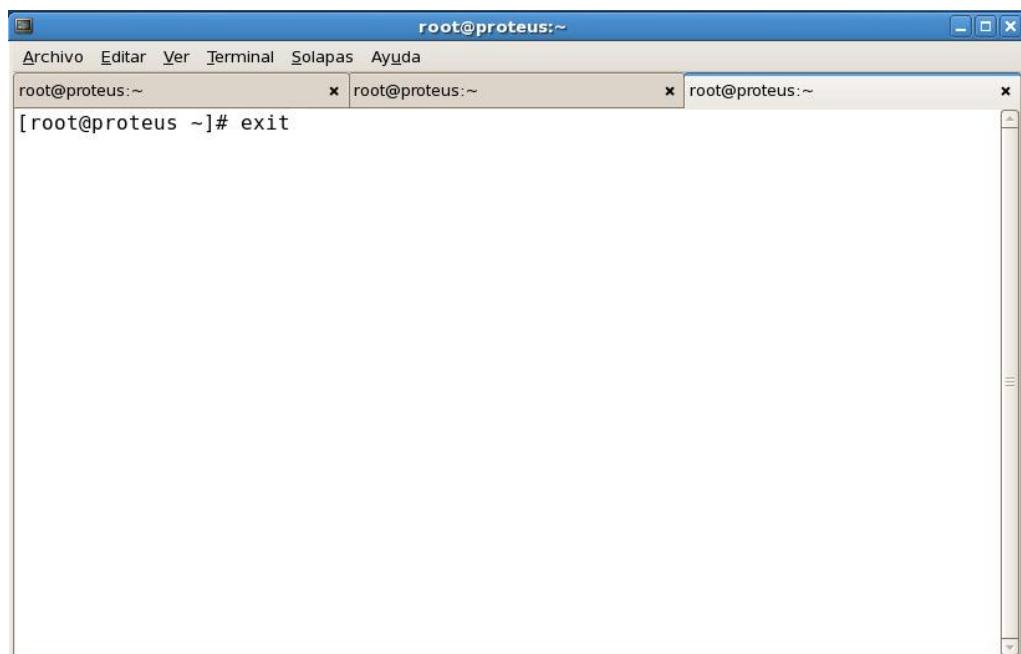


Figura 2.5. Cerrar Sesión con el comando exit

2.1.3. Cerrar la sesión gráfica

Para salir del Sistema GNU/Linux, dar clic en la barra de tareas (parte superior del escritorio) en el botón **Sistema** y del menú seleccione **Salir**. Cuando el diálogo de confirmación aparece como se muestra en la Figura 2.6, presione el botón **Salir**.



Figura 2.6. Confirmación de la salida

2.1.4. Apagar su ordenador

Antes de apagar su ordenador, es importante que cierre apropiadamente CentOS.

a. Cerrar en modo gráfico

Del menú en la barra de tareas, seleccione Sistema y luego seleccione la opción Apagar. Luego pedirá confirmación presione el botón Apagar.



Figura 2.7. Confirmación para apagar el sistema

Algunos ordenadores desconectan el poder automáticamente después de cerrar CentOS. Si su ordenador no lo hace, puede apagar su equipo con seguridad después que vea el mensaje:

```
Power down.
```

b. Reiniciar el Sistema

Para reiniciar su computador desde el intérprete de comandos, escriba el comando siguiente:

```
reboot
```

```
o
```

```
shutdown -r now
```

c. Cerrar o apagar el sistema

Para apagar su computador desde el intérprete de comandos, escriba el siguiente comando:

```
halt
```

```
o
```

```
shutdown -h now
```

Algunos ordenadores se apagan automáticamente después de cerrar el sistema CentOS. Si su ordenador no lo hace, puede apagar su equipo con seguridad después que vea el mensaje: **System halted.**

2.1.5. Propiedades del terminal

Cuando abre un terminal, ésta muestra las siguientes propiedades. Ver Figura 2.8.

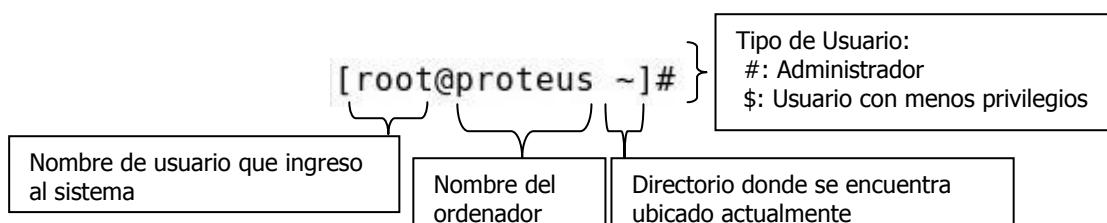


Figura 2.8. Prompt del Terminal

2.2. Estructura del árbol de directorios

Debido a la gran cantidad de distribuciones GNU/Linux existentes (slackware, mandriva, suse, ubuntu, etc.), se ha creído conveniente normalizar la estructura de directorios con la finalidad de encontrar los archivos sin importar la distribución instalada. CentOS y otras distribuciones utilizan el estándar FHS (Filesystem Hierarchy Standard).

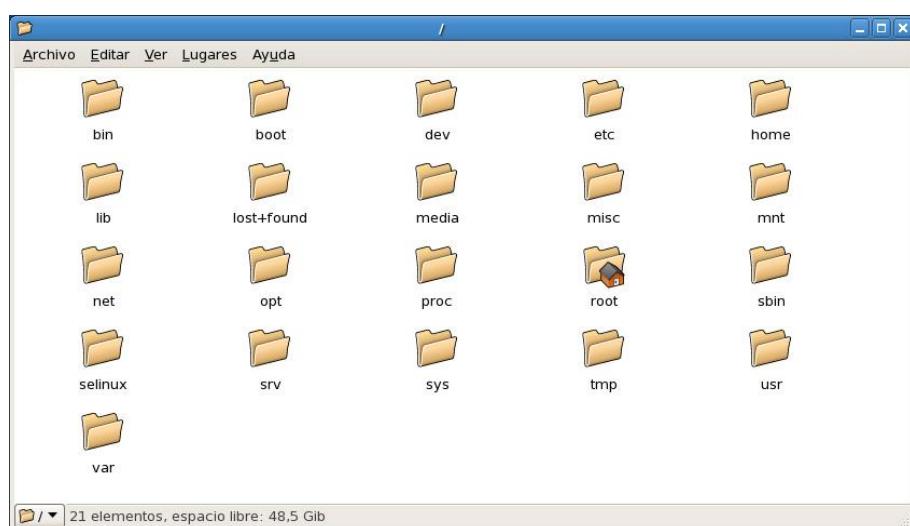


Figura 2.9. Raíz del Sistema GNU/Linux

2.2.1. Directorio /bin

Contiene los programas ejecutables disponibles para los usuarios como: cat, cp, ls, clear, more, less, tar, entre otros.

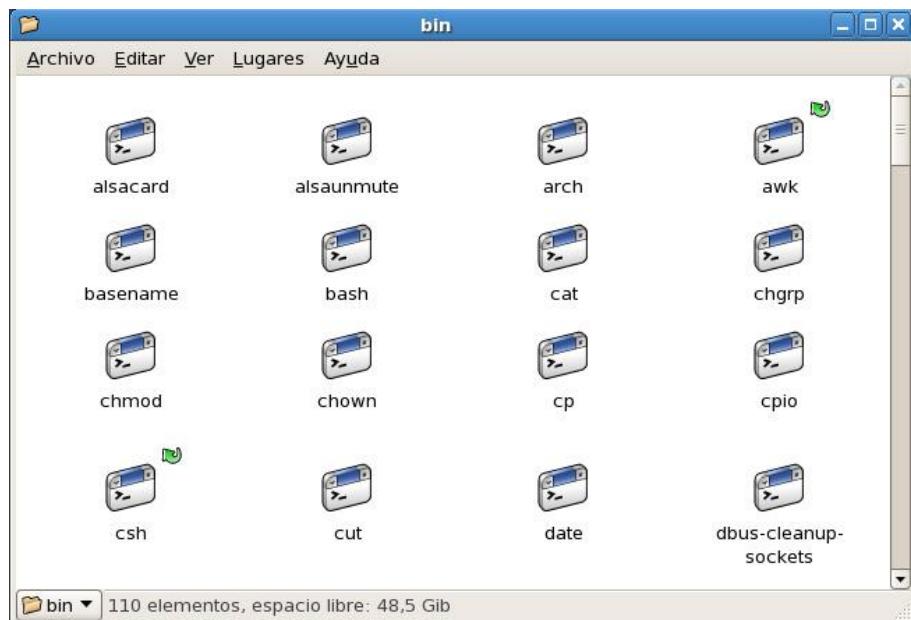


Figura 2.10. Directorio bin

2.2.2. Directorio /boot

Directorio del booteo (donde reside el Kernel de Linux) y algunos archivos necesarios para la inicialización del Sistema Operativo GNU/Linux. En ella se guarda la configuración del gestor de arranque.



Figura 2.11. Directorio boot

2.2.3. Directorio /dev

Contiene todos los archivos de dispositivos del sistema. GNU/Linux trata cada dispositivo (terminales, discos, impresoras, interfaz de red, etc.) como un archivo (tty1, hda, lp0, eth0, etc.).

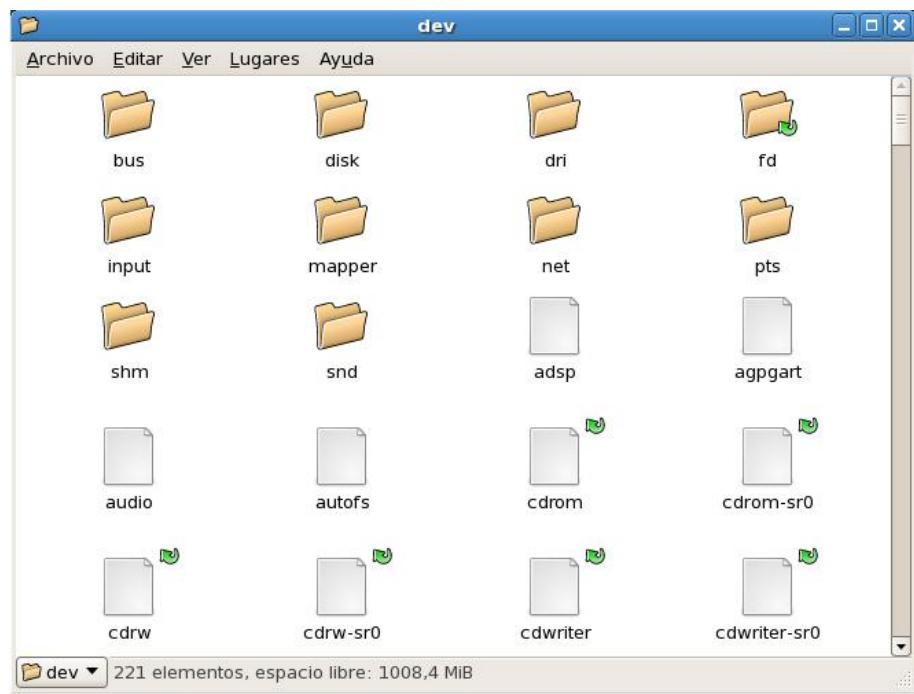


Figura 2.12. Directorio dev

2.2.4. Directorio /etc

Contiene los archivos de configuración del sistema y los servicios que se ha instalado en el sistema. Dentro de este directorio se encuentran dos subdirectorios: `skel` (skeleton) que sirve como archivos esqueletos que copiados al directorio personal del usuario cuando este se crea y el subdirectorio `X11` que son los archivos de configuración del sistema X-Window. Además de los archivos de configuración de interfaz de red, usuarios, grupos y claves de los usuarios, entre otros.

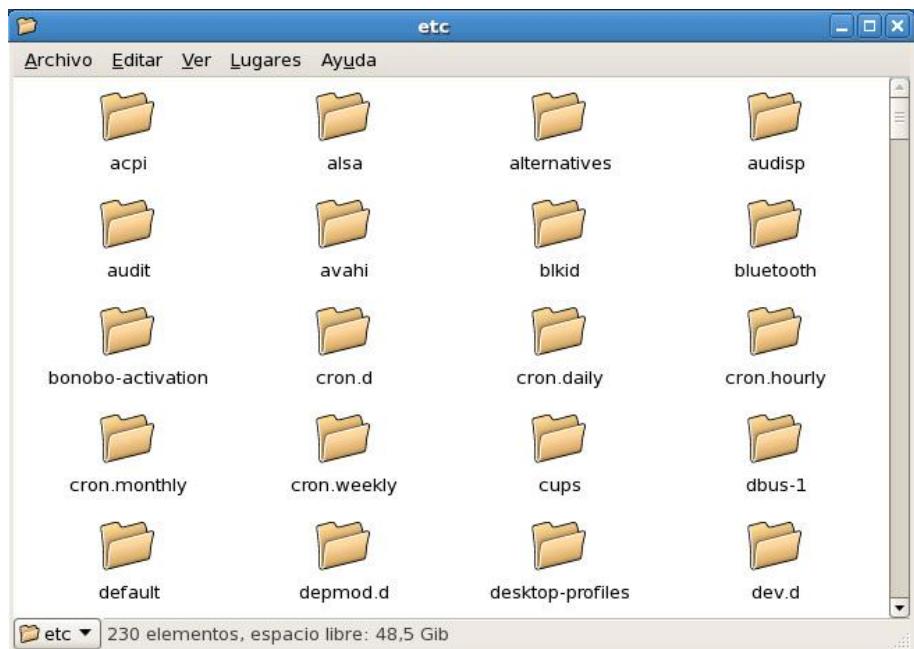


Figura 2.13. Directorio etc

2.2.5. Directorio /home

Contiene los directorios personales de los usuarios. El directorio HOME (directorio base) es el directorio inicial donde será posicionado un usuario al ingresar al sistema.



Figura 2.14. Directorio home

2.2.6. Directorio /lib

Directorio que contiene librerías estáticas y dinámicas que se necesitan para ejecutar las aplicaciones y utilidades del sistema. Además contiene los módulos necesarios del Kernel.

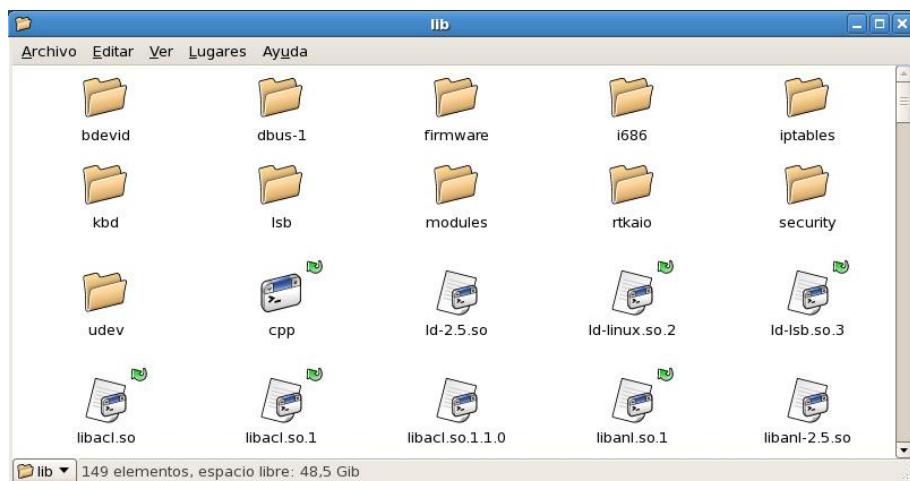


Figura 2.15. Directorio lib

2.2.7. Directorio /lost+found

Directorio para archivos recuperados por el proceso de reparación del sistema de archivos, que se ejecuta luego de una caída del sistema y asegura su integridad luego de que el equipo haya sido apagado de manera inapropiada.



Figura 2.16. Directorio lost+found

2.2.8. Directorio /media

Contiene los subdirectorios utilizados como punto de montaje de los dispositivos removibles (uso temporal) tales como CDROM, discuetes, usb, etc.

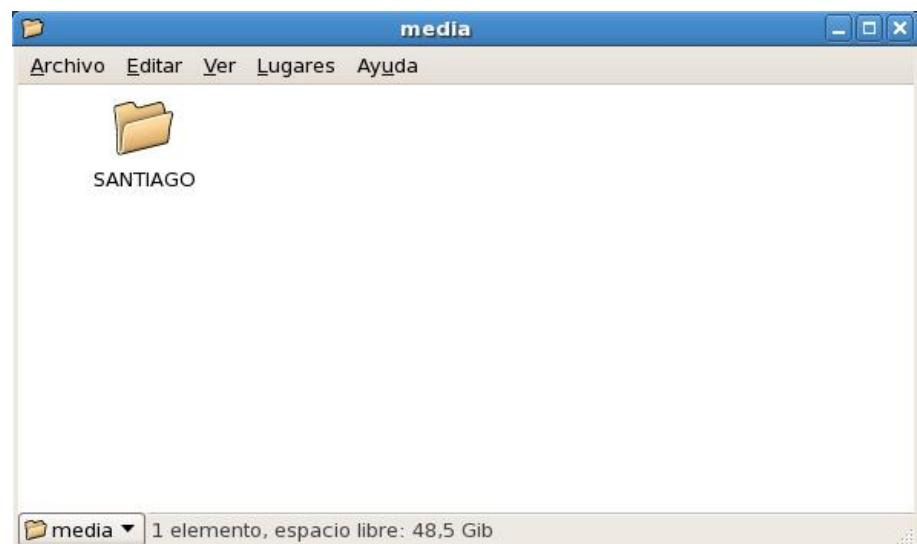


Figura 2.17. Directorio media

2.2.9. Directorio /misc

Abreviación de miscelánea. Utilizado para propósitos variados.



Figura 2.18. Directorio misc

2.2.10. Directorio /mnt

Reservado para sistemas de archivos montados temporalmente, tales como los montajes de NFS (Network Files System). Para los dispositivos de uso temporal (USB, CDROM) utilizar el directorio /media.



Figura 2.19. Directorio mnt

2.2.11. Directorio /net

Utilizado como punto de montaje para sistemas de archivos remotos. Por ejemplo NFS (Network Files System).



Figura 2.20. Directorio net

2.2.12. Directorio /opt

Es un espacio reservado para almacenar paquetes de terceros que no están incluidos en la misma distribución. Ejemplo: oracle, staroffice, etc.

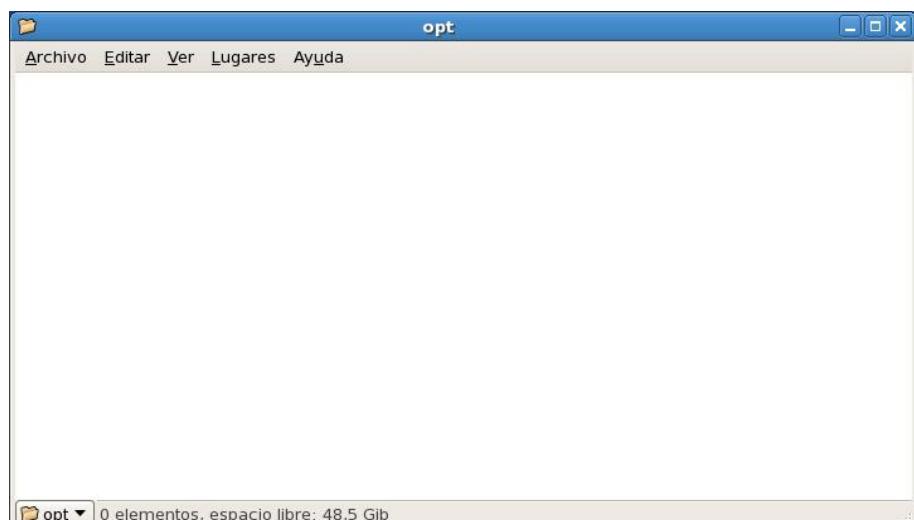


Figura 2.21. Directorio opt

2.2.13. Directorio /proc

Contiene archivos con información sobre el estado de ejecución del sistema operativo y de los procesos. Esta información es almacenada en tiempo real y creada en la memoria virtual del sistema.

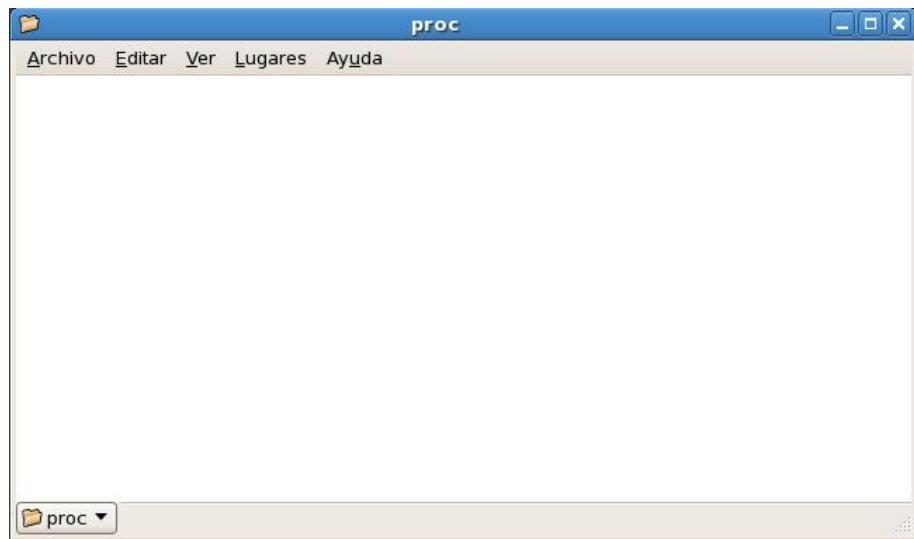


Figura 2.22. Directorio proc

2.2.14. Directorio /root

Es el directorio personal (HOME) para el usuario root (administrador del sistema GNU/Linux).



Figura 2.23. Directorio root

2.2.15. Directorio /sbin

Contienen archivos ejecutables de administración que son usados solamente por el usuario root (administrador del sistema) para el mantenimiento del sistema. Por ejemplo: ifconfig, iptables, etc.



Figura 2.24. Directorio sbin

2.2.16. Directorio /selinux

El pseudo-sistema de archivos /selinux contiene los comandos que son utilizados más a menudo por el subsistema del kernel. Este tipo de sistema de archivos es similar al pseudo sistema /proc.

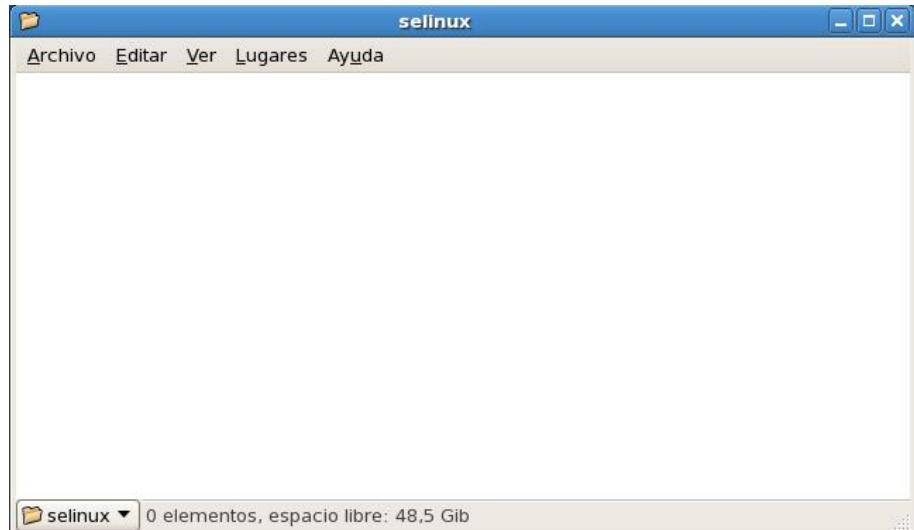


Figura 2.25. Directorio selinux

2.2.17. Directorio /srv

Contiene datos específicos de los oficios ofrecidos por el sistema.



Figura 2.26. Directorio srv

2.2.18. Directorio /sys

Sistema de archivos virtual sysfs específico del kernel 2.6. Este directorio contiene información similar a la que se encuentra en /proc, pero muestra una vista jerárquica de la información de dispositivos específicos con relación a los dispositivos de conexión en caliente.



Figura 2.27. Directorio sys

2.2.19. Directorio /tmp

Directorio donde se almacenan los archivos temporales del sistema.



Figura 2.28. Directorio tmp

2.2.20. Directorio /usr

Contiene archivos de programas, de datos y librerías asociados con las actividades de los usuarios.

Contiene todos los programas para usuarios (/usr/bin), bibliotecas /usr/lib), documentación (/usr/share/doc), etc. Dispone de sus propia estructura jerárquica interna y solamente es posible compartirlo en modo lectura. Es uno de los espacios que requiere de mayor espacio.



Figura 2.29. Directorio usr

2.2.21. Directorio /var

Este directorio contiene archivos de datos que representan información variable. Archivos de registro del sistema, tales como messages y lastlog que se ubican en el directorio /var/log. El directorio /var/lib/rpm contiene la base de datos de los paquetes RPM. Los archivos lock se ubican dentro de /var/lock, habitualmente en directorios para el programa usando el archivo. El directorio /var/spool contiene subdirectorios para programas en los que se almacenan archivos de datos, por ejemplo mail.



Figura 2.30. Directorio var

Fuente: Red Hat, Inc (2005) "Red Hat Enterprise Linux 4 - Introducción a la administración de sistemas" publicado en <http://www.redhat.com/docs/manuals/enterprise/RHEL-4-Manual/es/ref-guide/>"

Lección 3

Comandos Básicos

3.1. Comando date

Este comando tiene dos funciones: la primera mostrar en pantalla la fecha del sistema y segundo configurar la hora del sistema, pero para que esta funcionalidad se cumpla, se debe ejecutar el comando como usuario root (administrador del sistema).

Sintaxis: date [MMDDhhmmYY][.ss]]

Donde: MM=mes, DD=día, hh=hora, mm=minuto, YY=año, ss=segundos

Ejemplos:

```
[root@fisct ~]# date  
sáb ene 16 13:31:04 PET 2010 # Nos muestra la fecha y hora actual
```

Para establecer entonces la fecha al 20 de Julio del 2010 a las 8:05 de la noche:

```
[root@fisct ~]# date 0720200510  
mie jul 20 20:05:04 PET 2010 # Nos muestra la nueva fecha y hora
```

Nota: El comando **date**, como se ha descrito, establece la fecha y hora del sistema, que es diferente a la fecha y hora del hardware o la BIOS. Esta fecha del reloj físico del sistema se puede consultar utilizando el comando **hwclock**. Si desea que la fecha del sistema sea igual a la del hardware, o la del hardware igual a la fecha del sistema, utilizar las siguientes opciones:

```
[root@fisct ~]# hwclock --hctosys # reloj hardware a reloj sistema  
[root@fisct ~]# hwclock --systohc # reloj sistema a reloj hardware
```

3.2. Comando cd

El comando **cd** (change directory) se utiliza para cambiar el directorio actual.

Sintaxis: cd [directorio]

Ejemplos:

```
# cd /tmp      # Cambia al directorio tmp  
# cd          # Cambia hacia el directorio base (home directory) del usuario  
               # actual (si usted ingreso como usuario root regresará a su  
               # directorio base (/root))  
# cd /usr/local # Cambia al directorio local que se encuentra ubicado dentro el  
                 # directorio /usr  
# cd -        # Similar al uso del comando cd  
# cd ..       # Permite retroceder a un directorio anterior  
# cd ~        # Cambia hacia el directorio base del usuario actual. El carácter ~  
               # bash lo interpreta como el directorio base del usuario que  
               # ingreso al sistema  
# cd ~root    # Cambia hacia el directorio base del usuario root
```

3.3. Comando clear

Borrar de la ventana del Terminal la secuencia de comandos ejecutados o los resultados mostrados en pantalla.

Sintaxis: clear

Ejemplo:

```
[root@fiscct ~]# clear
```

3.4. Comando pwd

Para mostrar el directorio o posición actual en el sistema GNU/Linux se utiliza el comando **pwd** (printing working directory), que mostrará la ruta completa del directorio en el cual se encuentra ubicado.

Sintaxis: pwd

Ejemplo:

```
[root@fiscct ~]# cd  
[root@fiscct ~]# pwd  
/root # Muestra en pantalla la posición actual
```

```
[root@fiscct ~]# cd /usr/local  
[root@fiscct ~]# pwd  
/usr/local # Muestra en pantalla la posición actual
```

3.5. Listar ficheros (Archivos y Directorios)

- Comando ls

Uno de los comandos mas utilizados es **ls** que permite listar ficheros (archivos y directorios).

Sintaxis: ls [opciones] [fichero(s)...]

Si ejecuta **ls** sin argumentos, dará como resultado un listado de todos los ficheros (archivos y directorios) del directorio donde el usuario está posicionado. Para consultar el directorio donde está posicionado ejecute el comando **pwd**.

Las opciones del comando **ls** son:

- l** Lista los ficheros con mucho mas detalle, especificando para cada fichero su permiso, el número de enlaces rígidos, el nombre del propietario, el grupo al que pertenece, el tamaño en bytes, y la fecha de modificación.
- a** Lista todos los ficheros, incluyendo aquellos que comienzan con un «.» que representa a los ficheros ocultos.
- r** Invierte el orden de listado de los ficheros.
- s** Muestra el tamaño de cada fichero en bloques de 1024 bytes a la izquierda del nombre.
- t** Lista los ficheros ordenados por el tiempo de modificación en vez de ordenarlos alfabéticamente.
- A** Lista todos los ficheros (ocultos y no ocultos) excepto el «.» y el «..».

- R** Lista los contenidos de todos los directorios recursivamente.
- S** Ordena el listado por el tamaño de los ficheros.
- F** Si se ejecuta el comando ls con la opción -F, mostrará una lista de los ficheros marcados con un símbolo que indica la clase o tipo de fichero.

Ejemplos:

Ejecute la siguiente línea de comando:

```
[root@fisct ~]# cd
```

```
[root@fisct ~]# ls -l
```

```
-rw----- 1 root root 958 dic 30 05:04 anaconda-ks.cfg
drwxr-xr-x 2 root root 4096 dic 30 05:12 Desktop
-rw-r--r-- 1 root root 36613 dic 30 05:04 install.log
-rw-r--r-- 1 root root 4425 dic 30 05:01 install.log.syslog
```

Muestra un listado con información de los ficheros donde cada columna
representa a:

1er campo: Permisos
2do campo: Número de enlaces
3er campo: Dueño del fichero
4to campo: Grupo al que pertenece el fichero
5to campo: Tamaño en bytes
6to campo: Fecha y hora de creación o modificación del fichero
7mo campo: Nombre del fichero

```
[root@fisct ~]# ls -a
```

```
.           .bashrc   .esd_auth      .gstreamer-0.10    .metacity
..          .cshrc    .gconf        .gtkrc-1.2-gnome2  .mozilla
anaconda-ks.cfg Desktop  .gconfd       .ICEauthority   .nautilus
.bash_history   .dmrc     .gnome       install.log     .redhat
.bash_logout    .eggcups   .gnome2      install.log.syslog .tcshrc
.bash_profile   .elinks   .gnome2_private .lessht       .Trash
```

Muestra un listado de los ficheros ocultos (empiezan con punto (.)) y no ocultos

```
[root@fisct ~]# ls -F
```

```
anaconda-ks.cfg Desktop/ install.log install.log.syslog
```

Indica el tipo de fichero al final de cada fichero (/=directorio, *=ejecutable,
@=enlace simbólico)

```
[root@fisct ~]# ls -la
```

```
drwxr-x--- 16 root root 4096 ene 19 11:49 .
drwxr-xr-x 24 root root 4096 ene 19 08:47 ..
-rw----- 1 root root 958 dic 30 05:04 anaconda-ks.cfg
-rw----- 1 root root 5828 ene 20 15:42 .bash_history
-rw-r--r-- 1 root root 24 ene 6 2007 .bash_logout
-rw-r--r-- 1 root root 191 ene 6 2007 .bash_profile
-rw-r--r-- 1 root root 176 ene 6 2007 .bashrc
-rw-r--r-- 1 root root 100 ene 6 2007 .cshrc
drwxr-xr-x 2 root root 4096 dic 30 05:12 Desktop
-rw----- 1 root root 26 dic 30 05:12 .dmrc
drwxr-x--- 2 root root 4096 dic 30 05:12 .eggcups
drwx----- 2 root root 4096 ene 12 11:32 .elinks
-rw----- 1 root root 16 dic 31 10:07 .esd_auth
drwx----- 5 root root 4096 ene 19 08:48 .gconf
drwx----- 2 root root 4096 ene 19 08:49 .gconfd
drwxr-xr-x 3 root root 4096 dic 30 05:12 .gnome
drwx----- 6 root root 4096 ene 19 08:49 .gnome2
drwx----- 2 root root 4096 dic 30 05:12 .gnome2_private
drwxr-xr-x 2 root root 4096 dic 30 09:56 .gstreamer-0.10
-rw-r--r-- 1 root root 81 dic 30 05:12 .gtkrc-1.2-gnome2
-rw----- 1 root root 1687 ene 19 08:49 .ICEauthority
-rw-r--r-- 1 root root 36613 dic 30 05:04 install.log
-rw-r--r-- 1 root root 4425 dic 30 05:01 install.log.syslog
-rw----- 1 root root 35 ene 19 12:20 .lessht
drwx----- 3 root root 4096 dic 30 05:12 .metacity
drwx----- 4 root root 4096 dic 30 06:00 .mozilla
drwxr-xr-x 3 root root 4096 ene 19 08:49 .nautilus
drwxr-xr-x 3 root root 4096 dic 30 05:12 .redhat
-rw-r--r-- 1 root root 129 ene 6 2007 .tcshrc
```

Muestra un listado de la información de los ficheros ocultos y no ocultos

```
[root@fisct ~]# ls -ls
```

```
8 -rw----- 1 root root 958 dic 30 05:04 anaconda-ks.cfg
4 drwxr-xr-x 2 root root 4096 dic 30 05:12 Desktop
40 -rw-r--r-- 1 root root 36613 dic 30 05:04 install.log
12 -rw-r--r-- 1 root root 4425 dic 30 05:01 install.log.syslog
```

Muestra un listado de los ficheros con el tamaño de los ficheros en la primera
columna

```
[root@fisct ~]# ls -IS
```

```
-rw-r--r-- 1 root root 36613 dic 30 05:04 install.log
-rw-r--r-- 1 root root 4425 dic 30 05:01 install.log.syslog
drwxr-xr-x 2 root root 4096 dic 30 05:12 Desktop
-rw----- 1 root root 958 dic 30 05:04 anaconda-ks.cfg
```

Muestra un listado de los ficheros ordenado por el tamaño

```
[root@fisct ~]# ls -lt
```

```
drwxr-xr-x 2 root root 4096 dic 30 05:12 Desktop
-rw----- 1 root root 958 dic 30 05:04 anaconda-ks.cfg
-rw-r--r-- 1 root root 36613 dic 30 05:04 install.log
-rw-r--r-- 1 root root 4425 dic 30 05:01 install.log.syslog
```

Muestra un listado de los ficheros ordenado por el tamaño

```
[root@fisct ~]# ls -lr
-rw-r--r-- 1 root root 4425 dic 30 05:01 install.log.syslog
-rw-r--r-- 1 root root 36613 dic 30 05:04 install.log
drwxr-xr-x 2 root root 4096 dic 30 05:12 Desktop
-rw----- 1 root root 958 dic 30 05:04 anaconda-ks.cfg

# Muestra un listado de los ficheros ordenado de forma descendente

[root@fisct ~]# ls -lrt
-rw-r--r-- 1 root root 4425 dic 30 05:01 install.log.syslog
-rw-r--r-- 1 root root 36613 dic 30 05:04 install.log
-rw----- 1 root root 958 dic 30 05:04 anaconda-ks.cfg
drwxr-xr-x 2 root root 4096 dic 30 05:12 Desktop

# Muestra un listado de los ficheros ordenado por el tiempo de forma ascendente
```

3.6. Creación de Ficheros

3.6.1. Comando mkdir

El comando **mkdir** se utiliza para crear directorios.

Sintaxis: mkdir [argumento]

Ejemplos:

```
[root@fisct ~]# cd /opt
[root@fisct ~]# mkdir documentos
[root@fisct ~]# mkdir /opt/trabajos
[root@fisct ~]# mkdir -p docs/linuxdocs/ # con la opción se crean los directorios
# intermedios si es necesario
```

3.6.2. Comando touch

Este comando le permite crear archivos.

Sintaxis: touch [argumento]

Ejemplos:

```
[root@fisct ~]# cd /opt
[root@fisct ~]# touch ejemplo
[root@fisct ~]# touch ejemplo01 ejemplo02
[root@fisct ~]# touch "Plataforma Linux" # Permite crear el archivo con espacios
[root@fisct ~]# ls -l
```

3.7. Borrar Ficheros

Existen dos formas de borrar un fichero:

3.7.1. Comando rmdir

Para borrar un directorio (vacío) utilizar el comando **rmdir**.

Sintaxis: rmdir [directorio]

Ejemplo:

```
[root@fisct ~]# cd /opt
[root@fisct ~]# mkdir personal
[root@fisct ~]# rmdir personal
```

3.7.2. Comando rm

Permite borrar solo archivos.

Sintaxis: rm [opciones] [fichero(s)...]

Ejemplo:

```
[root@fisct ~]# cd /opt
[root@fisct ~]# touch file1 file2 file3 file4
[root@fisct ~]# rm file1
rm: ¿borrar el fichero regular vacío «file1»? (s/n) # Le solicitará
                                                    # confirmación
```

Si deseamos eliminar los ficheros sin que solicite la confirmación utilizar el comando **rm** con la opción **-r** (recursive) y **-f** (no pide confirmación)

```
[root@fisct ~]# rm -rf file2 file3 file4
```

Si deseamos eliminar un directorio que no está vacío, junto con los archivos y subdirectorios que contiene, utilizar el comando **rm** con la opción **-r** (recursive) y **-f** (no pide confirmación)

Ejemplo:

```
[root@fisct ~]# cd /opt
[root@fisct ~]# mkdir personal
[root@fisct ~]# cd personal
[root@fisct ~]# touch file1 file2 file3 file4
[root@fisct ~]# cd ..
[root@fisct ~]# rmdir personal
rmdir: personal: El directorio no está vacío
[root@fisct ~]# rm personal
rm: no se puede borrar el directorio «personal»: Es un directorio
```

Deberá ejecutar la siguiente orden:

```
[root@fisct ~]# rm -rf personal
```

3.8. Copiar Ficheros

- Comando cp

Este comando se utiliza para copiar ficheros.

Sintaxis: cp [opciones] fichero-origen... directorio-destino

Entre las opciones más relevantes:

- f Borrar los archivos de destino ya existentes.
- p Preservar los permisos, el usuario y el grupo del archivo a copiar.
- R Copia directorios recursivamente.
- v Da información en pantalla sobre los archivos que se van copiando.

Ejemplo:

Crear los siguientes ficheros:

```
[root@fisct ~]# cd /opt  
[root@fisct ~]# mkdir personal  
[root@fisct ~]# cd personal  
[root@fisct ~]# touch file1 file2 file3 file4 file5  
[root@fisct ~]# mkdir documentos
```

Copiar los ficheros file1 y file2 a documentos

```
[root@fisct ~]# cp file1 file2 documentos  
[root@fisct ~]# ls documentos  
file1 file2
```

En el siguiente caso cuando el directorio-destino no existe veamos lo que sucede.

```
[root@fisct ~]# cp file3 trabajos  
[root@fisct ~]# ls  
documentos file1 file2 file3 file4 file5 trabajos      # Al no estar creado el  
                                                        # directorio-destino se crea  
                                                        # una copia del archivo file3  
                                                        # con el nombre de trabajos
```

3.9. Mover o renombrar ficheros

- Comando mv

Este comando se usa tanto para mover archivos, como para renombrarlos.

Sintaxis: mv [opción...] fichero-origen... directorio-destino

Si el último argumento, destino es un directorio existente, **mv** mueve cada uno de los otros archivos a destino. Algunas opciones de este comando son:

- f Borrar los archivos de destino existentes sin preguntar al usuario.
- v Muestra el nombre de cada archivo a ser movido.

Ejemplo:

```
[root@fisct ~]# cd /opt  
[root@fisct ~]# cd personal  
[root@fisct ~]# mv file3 file4 documentos  
[root@fisct ~]# ls  
documentos file1 file2 file5 trabajos      # los archivos file3 y file4 no  
                                                # se muestran en pantalla
```

```
[root@fisct ~]# ls documentos
file1 file2 file3 file4
# los archivos file3 y file4 han
# sido movidos a documentos
```

En el siguiente caso cuando el directorio-destino no existe veamos lo que sucede.

```
[root@fisct ~]# mv file5 expedientes
[root@fisct ~]# ls
documentos expedientes file1 file2 trabajos      # El archivo file5 fue
# renombrado como
# expedientes
```

3.10. Comando uname

Muestra en pantalla información acerca del sistema operativo.

Sintaxis: uname [opción...]

Ejemplo:

```
[root@fisct ~]# uname -a
Linux fisct.uigv.edu.pe 2.6.18-92.el5 #1 SMP Tue Jun 10 18:51:06 EDT 2008
x86_64 x86_64 x86_64 GNU/Linux
```

3.11. Comando man

Para obtener más información de un comando determinado puede hacer uso del comando **man**.

Sintaxis: man [comando]

Ejemplo:

```
[root@fisct ~]# man cp          # muestra información del comando cp
```

Así mismo, puede obtener información de un comando utilizando **--help**

```
[root@fisct ~]# cd --help
```

Lección 4

Editor de Texto VIM

VIM es el editor de texto más usado en las distintas distribuciones GNU/Linux y UNIX.

El comando utilizado es vi.

Sintaxis: vi <nombre_archivo>

El editor vi trabaja utilizando dos modos de edición:

- **Modo de Comando**

Cuando iniciamos con el editor vi éste está en modo de comandos

- **Modo de Inserción**

Cuando usamos el editor vi en el modo de inserción, añadimos o reemplazamos texto. Cuando se encuentra en modo de texto y quiere retornar a modo de comando, deberá presionar la tecla [ESC].

4.1. Insertar Texto

Para insertar texto deberá utilizar una serie de teclas, donde cada uno representa a un comando. Para cambiar de un comando a otro deberá presionar la tecla [ESC].

- **Comando i**

Coloca al vi en modo de inserción.

- **Comando I**

Coloca al vi en modo de inserción e inserta el texto al comienzo de la línea actual.

- **Comando a**

Coloca al vi en modo de inserción y comienza a añadir el texto después del cursor.

- **Comando A**

El cursor se ubica al final de la línea de texto.

- **Comando o**

Inserta una línea de texto debajo de la línea actual.

- **Comando O**

Inserta una línea de texto por encima de la línea actual.

4.2. Salir y grabar

Los siguientes comandos le permiten salir y grabar del editor vi. Se denominan comandos de línea. Para pasar al modo comando de línea deberá primero presionar la tecla [ESC] y luego digitar los dos puntos (:).

- **Comando :w ó :w!**

Este comando permite grabar los cambios que se han efectuado en el archivo.

- **Comando :w nombre_archivo**

Grabará el contenido del archivo que actualmente se está editando en un nuevo archivo. Si el archivo existe no dejará grabarlo y mostrará un mensaje de alerta.

- **Comando :w >> añadir_archivo**
Añade el contenido del archivo que actualmente se está editando al final del archivo añadir_archivo.
- **Comando :wq ó :wq!**
Permite grabar los cambios realizados y salir del editor vi.
- **Comando :q**
Este comando le permite salir del vi siempre y cuando no haya realizado ningún cambio.
- **Comando :q!**
Este comando le permite salir del editor vi sin grabar el archivo.

4.3. Personalizar el Entorno de Edición

- **Comando set**
 - :set number o :set nu
Precede a cada línea que se visualiza con su número de línea respectiva.
 - :set nonu
Deshabilita la enumeración de la línea de texto.
- El **comando G** permite ir al final del documento, o si se especifica un número de línea determinada, de esta forma, el comando 23G posiciona el cursor en la línea veintitrés y el comando 1G lo posiciona en la primera línea del archivo.

También puede utilizar el modo comando de línea para moverse a una línea específica. Solo bastará con pasar al modo comando de línea e ingresar el número en donde desea ubicarse. Así, el comando :1, moverá el cursor a la primera línea, el comando :14 moverá el cursor a la línea 14 y el comando :\$ moverá el cursor al final del archivo.

4.4. Borrando Texto

- **Comando x**
Borra el carácter donde se encuentra el cursor.
- **Comando X**
Borra el carácter antes del cursor.
- **Comando dw**
Borra las palabras siguientes.
- **Comando dd**
Borra la línea actual.

De igual forma agregando un número antes del comando hace que éste se ejecute varias veces. Por ejemplo 3x borra tres caracteres, 5dw borra cinco palabras y 8dd borra ocho líneas. Otros comandos para borrado de texto:

- **Comando d\$ ó comando D**
Borra desde el cursor hasta el final de la línea.

- **Comando d0**
Borra desde el cursor hasta el comienzo de la línea.

4.5. Anulación de Cambios y Eliminaciones

- **Comando u**
Anula el último cambio realizado (deshacer).
- **Comando U**
Recupera todos los cambios de una línea realizados desde la última vez que se movió a dicha línea.
- **Comando :e!**
Este comando desecha todos los cambios realizados desde la última vez que se grabó el archivo. Recuerde que los dos puntos (:), indica modo de comando de línea.

4.6. Copiando y Moviendo Texto

Copiando texto

Para copiar emplearemos el comando y (por yank).

- **Comando y**
Este comando permite copiar porciones de texto al buffer de la memoria. Esto no elimina el texto original.

yw	Copia una palabra al buffer
y\$	Copia al buffer el texto desde la posición actual del cursor al final de la línea
yy o Y	Copia una línea al buffer
3yw	Copia tres palabras al buffer
2yy	Copia dos líneas al buffer

Para recuperar el texto que ha copiado, debe de ubicarse en la posición deseada y emplear el comando p o el comando P.

Moviendo texto

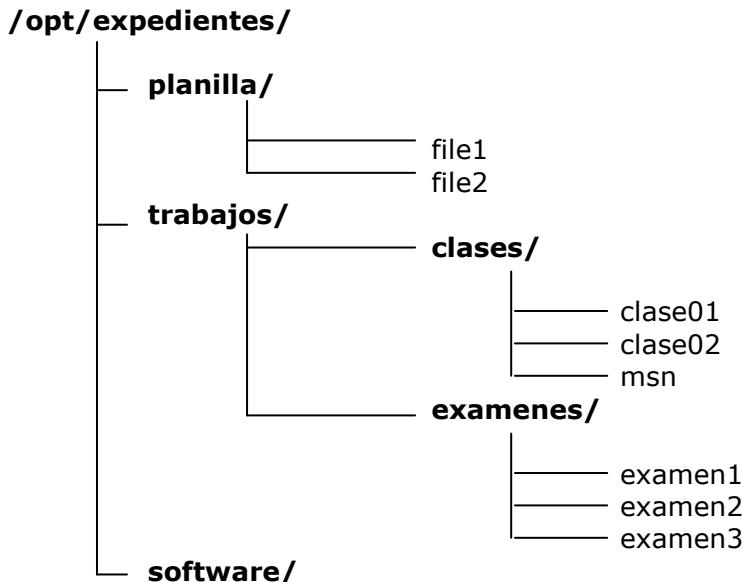
Para mover un texto, primero debe de eliminar el texto, luego colocarlo en la ubicación que desea.

Cuando se elimina algún texto, el vi coloca este material en el buffer de trabajo. Si se borra más texto, este buffer se sobrescribirá de manera que siempre contiene el material recientemente eliminado.

- **Comando p**
Recupera el texto del buffer y lo coloca a la derecha del cursor. Si eliminó líneas completas, el texto se coloca debajo de la línea actual
- **Comando P**
Recupera el texto del buffer y lo coloca a la izquierda del cursor. Si se eliminó líneas completas, el texto se coloca encima de la línea actual.

Ejercicios resueltos

1. Defina la sintaxis para crear la siguiente estructura:



Para crear la estructura anterior deberá ingresar al directorio /opt

```
[root@fisct ~]# cd /opt
```

Crear el directorio expedientes

```
[root@fisct ~]# mkdir expedientes  
[root@fisct ~]# cd expedientes
```

Dentro del directorio expedientes, como indica la estructura anterior, se encuentra los directorios planilla, trabajos y software

```
[root@fisct ~]# mkdir planilla trabajos software
```

El directorio planilla contiene los archivos file1 y file2

```
[root@fisct ~]# touch planilla/file1  
[root@fisct ~]# touch planilla/file2
```

El directorio trabajos contiene los directorios clases y examenes, a su vez cada uno de ellos contiene archivos

```
[root@fisct ~]# mkdir trabajos/clases  
[root@fisct ~]# mkdir trabajos/examenes  
[root@fisct ~]# touch trabajos/clases01  
[root@fisct ~]# touch trabajos/clases02  
[root@fisct ~]# touch trabajos/msn  
[root@fisct ~]# touch trabajos/examenes/examen1  
[root@fisct ~]# touch trabajos/examenes/examen2  
[root@fisct ~]# touch trabajos/examenes/examen3
```

Por último crear la carpeta software

```
[root@fisct ~]# mkdir software
```

2. Describa la sintaxis para los siguientes casos

- 2.1 Copiar file1 a software.
- 2.2 Copiar file2 a clases.
- 2.3 Renombrar el fichero clase01 como clasemartes.
- 2.4 Mover la carpeta examenes a clases.
- 2.5 Mover el archivo msn al directorio software.
- 2.6 Eliminar el directorio clases.
- 2.7 Copiar la carpeta trabajos como trabajosrespaldo.
- 2.8 Eliminar el archivo clase02.

Solución:

- 2.1. Para copiar el archivo file1 al directorio software debe definir la ruta donde se encuentra, ubicándose dentro de la carpeta planilla.
[root@fisct ~]# cp planilla/file1 software
- 2.2. Para copiar el archivo file2 al directorio software debe definir la ruta donde se encuentra, ubicándose dentro de la carpeta planilla y clases ubicando dentro del directorio trabajos.
[root@fisct ~]# cp planilla/file2 trabajos/clases
- 2.3. Para renombrar el archivos clase01 debe asegurar que ningún fichero tenga el nombre clasemartes. Así mismo, la ruta de origen de clase01.
[root@fisct ~]# mv trabajos/clases/clase01 trabajos/clasemartes
- 2.4. Para mover el directorio examenes utilizaremos el comando mv. Deberá asegurarse que el directorio clases esté creado.
[root@fisct ~]# mv trabajos/examenes trabajos/clases
- 2.5. El archivo se msn se encuentra dentro del directorio clases y éste dentro del directorio trabajos.
[root@fisct ~]# mv trabajos/examenes/msn software
- 2.6. Para eliminar el directorio clases, utilizaremos el comando rm seguido de las opciones r y f.
[root@fisct ~]# rm -rf trabajos/clases
- 2.7. Para realizar una copia del directorio trabajos, deberá asegurarse que el fichero trabajosrespaldo no exista. Luego utilizar el comando cp seguido de las opciones r y f.
[root@fisct ~]# cp -rf trabajos trabajosrespaldo
- 2.8. El archivo clase02 se encuentra dentro del directorio clases y éste dentro del directorio trabajos. Se utilizará el comando rm seguido de las opciones r y f para saltar la confirmación de eliminación.
[root@fisct ~]# rm -rf trabajos/clases/clase02

3. Editor VIM

3.1. Con el comando vi, crear el siguiente archivo

```
[root@fisct ~]# cd /opt  
[root@fisct ~]# vi ejemplo
```

Al ingresar le mostrará lo siguiente:

Figura 4.1. Modo de comando al ingresar en el editor vi

3.2. Presione la tecla [ESC] y luego la tecla [i] para ingresar al modo de inserción

-- INSERT --

Figura 4.2. Modo de inserción

3.3. Estando en el modo de inserción, ingresar el siguiente texto:

Curso de Sistemas Operativos GNU/Linux

Linus Benedict Torvalds, estudiante de la Universidad Helsinki, lanzó la primera versión pública de su sistema operativo Linux. Unix es uno de los sistemas operativos más ampliamente difundido debido al soporte existente.

Curso de Sistemas Operativos GNU/Linux

Linus Benedict Torvalds, estudiante de la Universidad Helsinki, lanzo la primera version de su sistema operativo Linux
Unix es uno de los sistemas operativos ampliamente difundido debido al soporte existente.

1

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

~

- 3.6. Para copiar el primer párrafo deberá colocar el cursor al inicio del párrafo, luego presione dos veces la tecla [y]. Llevar el cursor al final del documento, para ello deberá presionar la tecla [ESC] seguido de la tecla [G]. Presione la tecla [p] para pegar el texto.

```
1 Curso de Sistemas Operativos GNU/Linux
2
3 Linus Benedict Torvalds, estudiante de la Universidad Helsinki, lanzo la primera
   version de su sistema operativo Linux
4 Unix es uno de los sistemas operativos ampliamente difundido debido al soporte ex
   istente.
5
6 Curso de Sistemas Operativos GNU/Linux
7
~  
~  
~  
~  
~  
~  
~  
~  
~  
~  
~  
-- INSERT --
```

- 3.7. Lleva el cursor al inicio del documento, para ello presionar la tecla [1][G].

```
1 Curso de Sistemas Operativos GNU/Linux
2
3 Linus Benedict Torvalds, estudiante de la Universidad Helsinki, lanzo la primera
   version de su sistema operativo Linux
4 Unix es uno de los sistemas operativos ampliamente difundido debido al soporte ex
   istente.
5
6 Curso de Sistemas Operativos GNU/Linux
7
~  
~  
~  
~  
~  
~  
~  
~  
~  
~
```

- 3.8. Para eliminar el primer párrafo, presionar dos veces la tecla [d].

```
1
2 Linus Benedict Torvalds, estudiante de la Universidad Helsinki, lanzo la primera
   version de su sistema operativo Linux
3 Unix es uno de los sistemas operativos ampliamente difundido debido al soporte ex
   istente.
4
5 Curso de Sistemas Operativos GNU/Linux
6
~  
~  
~  
~  
~  
~  
~  
~  
~  
~
```

- 3.9. Para salir y grabar del editor, presione la tecla [ESC] y escriba :wq!. Presiona la tecla [ENTER].

- 3.10. Al salir del editor regresará el prompt de línea de comandos.

[root@fisct ~]#

Lectura

Historia de UNIX

Unix es uno de los sistemas operativos ampliamente difundido, debido al soporte existente. La historia de Unix comienza a finales de los años 60, cuando en los Laboratorios Bell de AT&T y el fabricante de computadores GE (General Electric) trabajaron sobre un Sistema Operativo experimental denominado MULTICS. MULTICS (Multiplexed Information and Computing System – Información multiplexada y sistema de computación), fue diseñado como sistema operativo interactivo para la computadora GE 645, permitiendo la compartición de información al tiempo que proporcionaba seguridad. Estas empresas buscaban desarrollar “un gran sistema operativo interactivo” que incorporase sólidas políticas de seguridad. Hasta ese momento, la seguridad de los datos la proporcionaba la escasa disponibilidad de los ordenadores, de los que había una pequeña cantidad y solo personal autorizado podía tener acceso a ellos. Pero la baja de precios y la popularización de los sistemas informáticos hacían indispensable integrar en el mismo corazón del sistema operativo las herramientas destinadas a proteger la información.

Su desarrollo sufrió muchos retrasos, y las versiones de producción resultaron lentas y con grandes necesidades de memoria. Por una serie de razones, los laboratorios Bell abandonaron el proyecto, sin embargo, el sistema MULTICS implementó muchas características innovadoras y produjo un entorno de computación excelente.

En 1966, Ken Thompson, uno de los investigadores de los Laboratorios Bell involucrado en el proyecto MULTICS, escribió un juego para la computadora GE denominado Space Travel. Este juego simulaba el sistema solar y una nave espacial. Sin embargo, descubrió que el juego era lento en la máquina de General Electric y resultaba realmente caro, algo así como 75 dólares americanos por cada partida. Con la ayuda de Dennis Ritchie, Thompson volvió a escribir el juego para ejecutarse sobre un DEC PDP-7. Esta experiencia inicial le dio la oportunidad de escribir un nuevo sistema operativo sobre el PDP-7, utilizando la estructura de un sistema de archivos que habían diseñado Thompson, Ritchie y Rudd Canaday. Thompson, Ritchie y sus colegas crearon un sistema operativo multitarea, incluyendo un sistema de archivos, un intérprete de órdenes y algunas utilidades para el PDP-7.

Puesto que el nuevo sistema operativo multitarea para el PDP-7 podía soportar dos usuarios simultáneamente, se le denominó UNICS (Uniplexed Information and Computing – Información uniplexada y sistema de computación). El mismo Brian Kernighan fue el que eligió el nombre del nuevo sistema operativo, pero por culpa de un juego de palabras UNICS se convertía un sistema Multics castrado (pues “eunuchs”, en inglés, es un homófono de UNICS). Entonces, se decidió cambiarle el nombre a UNIX, denominación que se mantiene hasta la actualidad.

El Grupo de Investigación de Informática (Computer Science Research Group) quería seguir utilizando el Sistema UNIX, pero sobre una máquina más potente que el PDP-7. Ken Thompson y Dennis Ritchie gestionaron la obtención de un DEC PDP-11/20 en contrapartida a la promesa de añadir capacidades de procesamiento de texto al sistema UNIX. El sistema operativo UNIX, con el programa de formateado de texto runoff y un primitivo editor de texto, ambos escritos en lenguaje ensamblador, fueron portados al PDP-11/20 en 1970. Este sistema de procesamiento de texto inicial, sistema operativo UNIX, el editor y runoff, fueron adoptados por el departamento de patentes de los Laboratorios Bell como procesador de texto, runoff evolucionó a troff, el primer programa de edición electrónica con capacidad de composición tipográfica.

El 3 de noviembre de 1971, Thompson y Ritchie publicaron el primer manual de programación de UNIX, el “UNIX Programmer's Manual”.

En 1972, la segunda edición del manual del programador UNIX mencionaba que había exactamente diez computadoras utilizando el sistema UNIX. En 1973, Ritchie y Thompson volvieron a escribir el núcleo en un nuevo lenguaje de programación denominado "C", un lenguaje de alto nivel a diferencia de la mayor parte de los sistemas escritos para máquinas pequeñas que utilizaban generalmente un lenguaje ensamblador. La escritura del sistema operativo UNIX en C hacía mucho más fácil su mantenimiento y portabilidad a otras máquinas. La popularidad del sistema UNIX creció debido a sus innovaciones y a que podía modificarse de acuerdo a las preferencias individuales. AT&T puso UNIX a disposición de las universidades, empresas privadas y del gobierno de los Estados Unidos, a través de licencias. El Departamento de Computación de la Universidad de California, con sede en Berkeley recibió una de estas licencias, y en 1975 desarrolló y publicó su propio "clon" de UNIX, conocido como Berkeley Software Distribution (BSD), que más tarde se convertiría en un fuerte competidor del UNIX de AT&T. Para tener una idea de los alcances de UNIX en esa época, basta con una frase de junio de 1972 atribuida a Dennis Ritchie y Ken Thompson: "...el número de instalaciones UNIX ha alcanzado el número de 10, y esperamos que aumente..."

Los conceptos del Sistema UNIX continuaron creciendo, Los cauces, originalmente sugeridos por Doug McIlroy, fueron desarrollados por Ken Thompson al principio de los 70. La introducción de los cauces hizo posible el desarrollo de la filosofía UNIX, incluyendo el concepto de una caja de utilidades. Utilizando cauces, las utilidades se pueden conectar, tomando una entrada de otra utilidad y pasando la salida a una tercera.

Hacia 1974 comenzó a utilizarse ampliamente en los Laboratorios Bell la cuarta edición del sistema UNIX. Hacia 1975 salió la quinta y sexta edición, ésta última incluiría la denominada "pipes" (tuberías). El número de máquinas que ejecutan el sistema UNIX, fundamentalmente en los Laboratorios Bell y en las Universidades, se incrementó en más de 600 en 1978. La versión 7, última basada en el UNIX original que tuvo una gran distribución, entró en circulación en 1979 y sirvió de base para la creación de Plan 9, un nuevo sistema operativo portable y distribuido, diseñado por los Laboratorios Bell para ser el sucesor de UNIX en tareas de investigación.

La empresa AT&T desarrolló y vendió UNIX System III (basado en la versión 7) a partir de 1981. La proliferación de versiones daba lugar a confusiones, así que la empresa decidió combinar todos los desarrollos propios con los de distintas universidades y empresas en 1983, dando origen al Unix System V Release 1. Esta versión introdujo características como el editor vi y la biblioteca curses, desarrolladas por Berkeley Software Distribution. La división Unix Systems Laboratories de AT&T fue adquirida por Novell dos años más tarde, y se hizo cargo de la demanda por infracción de los derechos de copyright, revelación de secretos y violación de marca de mercado existente entre Unix Systems Laboratories y BSD. Los accionistas de Novell tuvieron que pasar el mal trago de descubrir grandes porciones del código de BSD habían sido copiado ilegalmente en UNIX System V, y fueron contra demandados. Como la propiedad intelectual de Novell se reducía a unos pocos ficheros fuente, todo acabó en un acuerdo extrajudicial cuyos términos permanecieron bajo secreto a petición de Novell.

De forma paralela al UNIX, desde mediados de los ochenta, Richard Stallman, del Instituto Tecnológico de Massachusetts, trabajaba en lo que más tarde se conocería como "software libre". Stallman creó un sistema similar a UNIX con intenciones de cederlo gratuitamente, con el nombre de GNU (Gnu's Not Unix. GNU no es Unix).

Fuente [5].

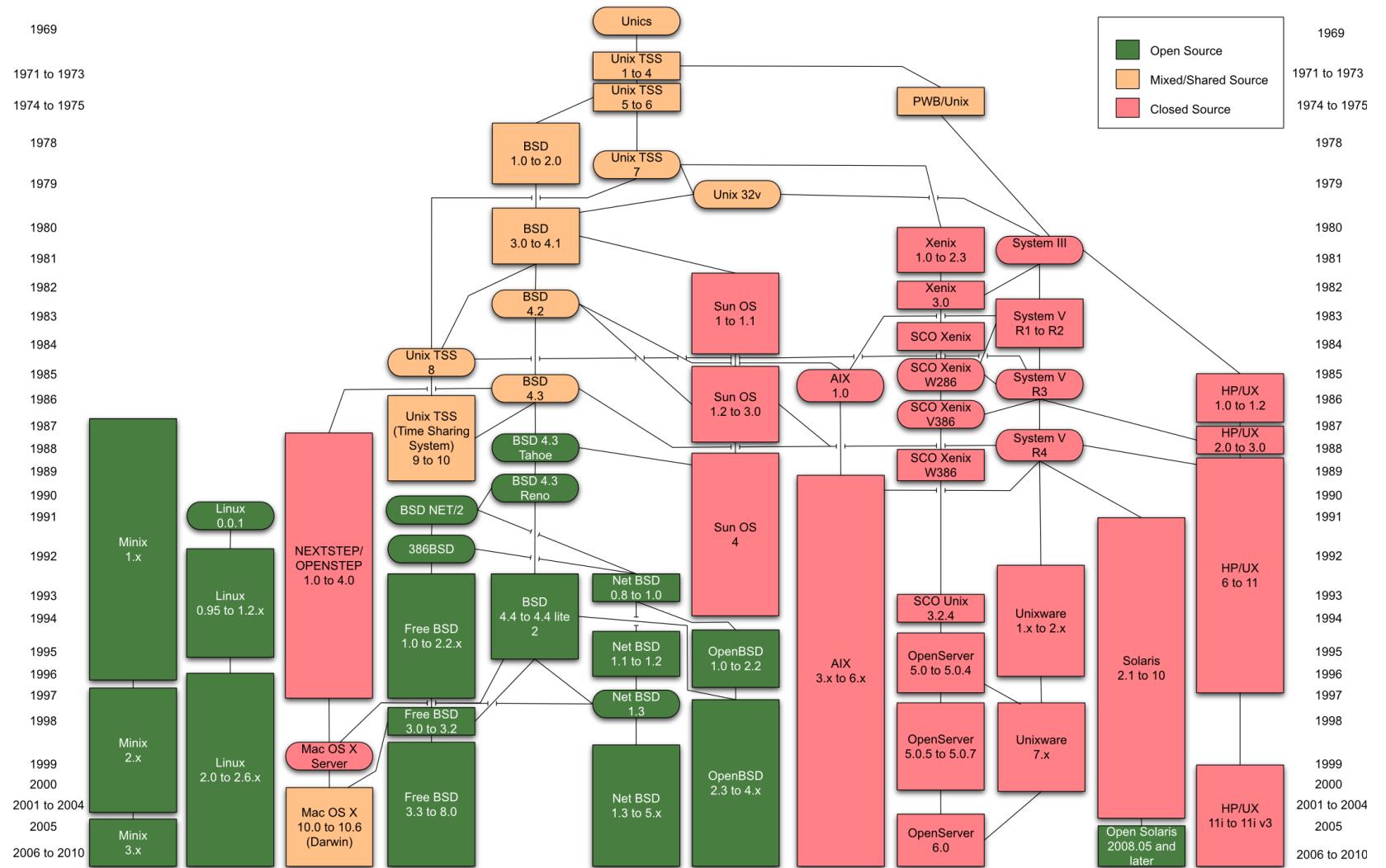


Figura 01. Evolución de UNIX

Fuente: <http://www.levenez.com/unix/>. Licencia Creative Commons Genérica de Atribución/Compartir-Igual 3.0.

Resumen

Esta unidad contiene los conceptos sobre la historia de UNIX y GNU/Linux, así como definiciones básicas de los comandos necesarios para trabajar en el sistema GNU/Linux, se muestran ejemplos prácticos para su ejecución.

Así mismo, se define la estructura del árbol de directorios y el uso del editor de texto VIM, utilizado en la mayoría de los sistemas UNIX y GNU/Linux.

Autoevaluación

I. Marcar la respuesta correcta.

1.1 Es el encargado de ser el intérprete de comandos ejecutados por el usuario:

- a. Kernel b. root c. grub d. bash e. tty1 f. su -

1.2 Corresponde a un dispositivo de disco extraíble:

- a. lp0 b. hda1 c. eth0 d. ttyS0 e. fd0 f. pts/1

1.3 Corresponde a un dispositivo de terminal de texto:

- a. lp0 b. hda1 c. eth0 d. ttyS0 e. fd0 f. pts/1

1.4 Configura el inicio de dos o más Sistemas Operativos:

- a. boot b. pwd c. grub d. shadow e. bash f. root

1.5 Muestra el listado de ficheros por el tamaño (kilobytes o megabytes):

- a. ls -la b. ls -S c. ls -tal d. ls -li e. ls -lth f. ls -s

1.6 Muestra el listado de ficheros por el tiempo:

- a. ls -la b. ls -S c. ls -rl d. ls -li e. ls -lth f. ls -s

1.7 Para establecer la fecha al 24 de setiembre de 2007 a las 12:07 de la noche:

- a. date 2409120707 b. date 2409240707 c. date 0924070007
d. date 0924240707 e. date 0924000707 f. date 0924071207

1.8 Para establecer la fecha al 24 de octubre de 2008 a las 12:08 del mediodía:

- a. date 2410120808 b. date 1024120808 c. date 1024080008
d. date 1024240808 e. date 2410240808 f. date 1024081208

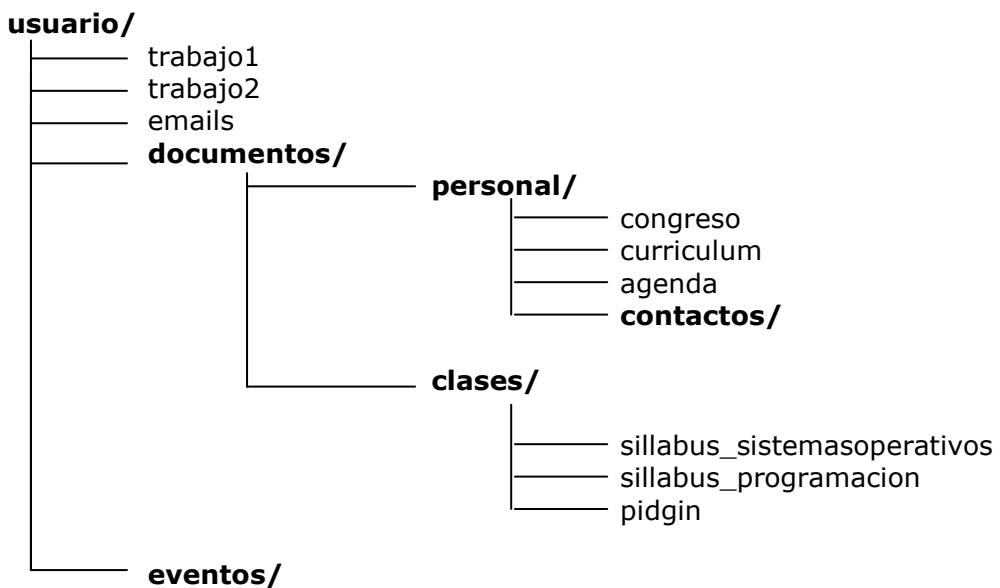
1.9 Opción utilizado por el comando uname para mostrar la versión del kernel:

- a. -m b. -r c. -v d. -i e. -o f. -p

1.10 ¿Cuál de los siguientes sistemas de archivos no puede realizarse un backup?

- a. home b. boot c. swap d. root e. proc f. etc

II. Defina la sintaxis para crear la siguiente estructura. El directorio usuario se creará dentro de la carpeta /opt:



Escriba la sintaxis correcta para realizar las siguientes tareas. Realizar todas las tareas desde el directorio usuarios.

- 2.1. Mover trabajo1 y trabajo2 a una carpeta llamado trabajos que estará dentro del directorio personal.
- 2.2. Mover emails al directorio a contactos.
- 2.3. Renombra clases como universidad.
- 2.4. Copiar congreso a eventos.
- 2.5. Eliminar el archivo agenda.
- 2.6. Mover pidgin a un directorio llamado software que está dentro de usuario.

Solucionario

1. Marcar la respuesta correcta:

- 1.1. d
- 1.2. e
- 1.3. f
- 1.4. c
- 1.5. e
- 1.6. e
- 1.7. e
- 1.8. b
- 1.9. b
- 1.10. c

2. Defina la sintaxis para crear la siguiente estructura:

```
[root@fisct ~]# mkdir usuario
[root@fisct ~]# cd usuario
[root@fisct ~]# touch trabajo1 trabajo2 emails
[root@fisct ~]# mkdir documentos eventos
[root@fisct ~]# mkdir documentos/personal documentos/clases
[root@fisct ~]# touch documentos/personal/congreso
[root@fisct ~]# touch documentos/personal/curriculum
[root@fisct ~]# touch documentos/personal/agenda
[root@fisct ~]# mkdir documentos/personal/contactos
[root@fisct ~]# touch documentos/clases/sillabus_sistemasoperativos
[root@fisct ~]# touch documentos/clases/sillabus_programacion
[root@fisct ~]# touch documentos/clases/pidgin
```

- 2.1. [root@fisct ~]# cd /opt/usuario
[root@fisct ~]# mkdir documentos/personal/trabajos
[root@fisct ~]# mv trabajo1 trabajo2 documentos/personal/trabajos
- 2.2. [root@fisct ~]# mv emails documentos/personal/contactos
- 2.3. [root@fisct ~]# mv documentos/clases documentos/universidad
- 2.4. [root@fisct ~]# cp documentos/personal/congreso eventos
- 2.5. [root@fisct ~]# rm -rf documentos/personal/agenda
- 2.6. [root@fisct ~]# mkdir software
[root@fisct ~]# mv documentos/universidad/pidgin software

Bibliografía

- [1]. Ball, Hill y Duff, Hoyt (2005) *Red Hat Linux. Fedora 3*. Madrid. Ediciones Anaya Multimedia.
- [2]. Bautts, Tony y Otros (2005) *Linux. Guía para Administradores*. Madrid. Ediciones Anaya Multimedia / O'Reilly.
- [3]. Kalle, Mathias y Welsh, Matt (2006) *Guía de Referencia y Aprendizaje LINUX*. 2º. Ed. Madrid, Ediciones Anaya Multimedia / O'Reilly.
- [4]. Negus, Christopher (2003) *Red Hat Linux 8*, Madrid. Ediciones Anaya Multimedia.
- [5]. Rosen, Kenneth y otros (1997) *Unix Sistema V, Versión 4*. Madrid. MCGRRAW-HILL / Interamericana de España, S.A. pp. 10-13.
- [6]. Von Hagen, Bill y Jones, Brian (2006) *Linux Server. Los mejores trucos*. Madrid. Anaya Multimedia/Wrox.

Enlaces

- Baig Viñas, Roger y Aulí Llinás (2003) "Sistema Operativo GNU/Linux Básico" *Formación de Posgrado de la UOC - Máster oficial de Software libre*.
http://www.uoc.edu/masters/oficiales/master_oficial_software_libre/master_oficial_software_libre_materiales.htm
- Kirch, Olaf y Dawson, Ferry (2002) "Guía de Administración de Redes con Linux". *O'Reilly (printed version) (c) 2000 O'Reilly & Associates. Proyecto LuCAS por la traducción al español*.
<http://es.tldp.org/Manuales-LuCAS/GARL2/garl2/>
- Red Hat, Inc (2005) "Red Hat Enterprise Linux 4 - Introducción a la administración de sistemas"
<http://www.redhat.com/docs/manuals/enterprise/RHEL-4-Manual/es/admin-guide/>

UNIDAD II

ADMINISTRANDO FICHEROS EN GNU/Linux

La unidad tiene como propósito que el estudiante conozca la administración de ficheros en los sistemas GNU/Linux, permitiéndole comprender su manejo y valorando la importancia de los conocimientos para su desarrollo académico. Contiene:

- Paginar y Visualizar ficheros
- Búsqueda de ficheros
- Filtrar Ficheros
- Empaquetar y comprimir Ficheros

Lección 5

Paginar, visualizar y búsqueda de ficheros

5.1. Paginar y Visualizar ficheros

5.1.1. Comando cat

El comando **cat** concatena (*catenate*) ficheros y los imprime en la salida estándar. Si no se le pasa ningún argumento lee de la entrada estándar. Existe también zcat que hace lo mismo, pero con ficheros compactados.

Este comando se utiliza para ver el contenido de un archivo en pantalla y sin pausa.

Sintaxis: cat <nombre_archivo>

Ejemplo:

```
[root@fisct ~]# cd /etc  
[root@fisct ~]# cat hosts  
  
# Do not remove the following line, or various programs  
# that require network functionality will fail.  
127.0.0.1      fisct.uigv.edu.pe localhost.localdomain localhost  
::1            localhost6.localdomain6 localhost6
```

Así mismo, podemos enviar la salida del comando **cat** a un archivo haciendo uso del símbolo **>**.

```
[root@fisct ~]# cat hosts > hosts.bak
```

Esta línea de comando copia el contenido del archivo hosts como hosts.bak. El símbolo mayor (**>**) proporciona una forma general para enviar la orden de una salida a un archivo.

En el ejemplo anterior, si no existe un archivo hosts.bak en el directorio actual (/etc), el sistema lo crea. Si ya existe un archivo con este nombre, la salida del comando **cat** lo sobrescribirá.

En ocasiones se quiere añadir información de un archivo en la parte final de otro. Para añadir información a un archivo existente, haga lo siguiente:

```
[root@fisct ~]# cat hosts >> hosts.bak
```

El símbolo **>>**, en el ejemplo anterior, añade el contenido del archivo denominado hosts al final del archivo denominado hosts.bak

5.1.2. Comando more

El comando **more** permite la visualización de un archivo por líneas o por pantalla. También existe la posibilidad de moverse hacia atrás o hacia delante y buscar patrones. El comando **more** permite controlar el porcentaje de visualización del archivo que se ha mostrado, de esta forma nos permitirá darnos cuenta si está por finalizar o no la visualización del archivo.

Si el archivo que está visualizando ocupa más de una pantalla, un prompt aparecerá en la parte inferior de cada pantalla mostrándose el porcentaje de visualización del archivo:

--More--(xx%)

En este prompt se puede continuar visualizando el contenido del archivo de varias formas:

- Presione la tecla <Barra Espaciadora> para mostrar la pantalla siguiente.
- Presione <Enter> para mostrar la siguiente línea.
- Digite un número seguido por "s" para saltar el número especificado de líneas.
- Presione la tecla "d" para moverse a media pantalla.
- Presione la tecla "b" para moverse una pantalla hacia atrás.
- Utilice la tecla slash (/) para buscar un texto en el archivo.
- Presione la tecla q para salir.

Sintaxis: more <nombre_archivo>

Ejemplo:

```
[root@fisct ~]# cd /etc  
[root@fisct ~]# more services
```

5.1.3. Comando less

Los comandos more y less paginan uno o varios archivos y los muestran en la terminal. De no indicársele un fichero, paginan la entrada estándar. Se diferencian en las facilidades que brindan, por ejemplo more es más restrictivo en cuanto al movimiento dentro del texto, mientras que less no limita este aspecto, pues acepta el empleo de todas las teclas de movimiento tradicionales. Cuando se alcanza el final del último fichero a paginar, more termina automáticamente, mas no sucede con el comando less. También more muestra sucesivamente el porcentaje del fichero visto hasta el momento.

Tanto less como more proveen una serie de comandos para moverse con facilidad dentro del texto paginado.

Sintaxis: less <nombre_archivo>

Ejemplo:

```
[root@fisct ~]# cd /etc  
[root@fisct ~]# less services
```

Se puede continuar visualizando el contenido del archivo de varias formas:

- Presione la tecla <Barra Espaciadora> para mostrar la pantalla siguiente.
- Presione <Enter> para mostrar la siguiente línea.
- Presione la tecla q para salir.

El comando **man**, para dar formato a su salida, utiliza por defecto el paginador less. Existen además los comando **zless** y **zmore** que permiten paginar con less y more respectivamente, a los ficheros compactados sin necesidad de descompactarlos previamente.

5.2. Búsqueda de ficheros

5.2.1. Comando grep

El comando grep es un programa de utilidad que busca en un archivo, o más de un archivo, líneas que contienen un cierto patrón.

Sintaxis: grep [opciones] <patron_busqueda> [archivo]

Ejemplo:

Supongamos que queremos encontrar si el nombre root se encuentra registrado en el archivo /etc/passwd:

```
[root@fisct ~]# grep root /etc/passwd
root:x:0:0:root:/root:/bin/bash
operator:x:11:0:operator:/root:/sbin/nologin
[root@fisct ~]#
```

5.2.2. Comando find

El comando find es un programa que puede buscar recursivamente a través de una estructura de directorios y hallar ficheros que satisfagan ciertos criterios. Esta orden trabaja con el criterio **name** para hallar los ficheros.

Sintaxis: find <ruta> [opciones] <patron_busqueda>

Ejemplo:

```
[root@fisct ~]# find / -name passwd
```

5.2.3. Comando locate

El comando locate busca en una base de datos, actualizada periódicamente, todos los *paths* en la jerarquía de ficheros que contengan una cadena determinada. Para crear esta base de datos o actualizarla se debe invocar por *root* el comando **updatedb** (o locate -u) que actualiza o registra los ficheros del sistema.

Sintaxis: locate <patron_busqueda>

Ejemplo:

```
[root@fisct ~]# updatedb
[root@fisct ~]# locate passwd
[root@fisct ~]# locate install.log
```

Lección 6

Filtrar Ficheros

6.1. Comando file

El comando file determina con cierto grado de precisión el tipo de un fichero que se le pasa como argumento.

Sintaxis: file <fichero>

Ejemplos:

```
[root@fisct ~]# file /etc/passwd  
/etc/passwd: ASCII text
```

```
[root@fisct ~]# file /usr/sbin/useradd  
/usr/sbin/useradd: ELF 32-bit LSB executable, Intel 80386, version 1 (SYSV), for  
GNU/Linux 2.6.9, dynamically linked (uses shared libs), for GNU/Linux 2.6.9,  
stripped
```

```
[root@fisct ~]# file /etc  
/etc: directory
```

6.2. Comando sort

El comando sort ordena las líneas de un fichero mostrándolas por la salida estándar. De no especificarse un fichero toma la entrada estándar.

Sintaxis: sort [opciones] [fichero]

Algunas opciones:

- r : ordena al revés.
- f : trata las mayúsculas y minúsculas por igual.

Ejemplo:

```
[root@fisct ~]# sort -f /etc/passwd
```

6.3. Comandos tail y head

Los comandos tail y head muestran respectivamente el final y el comienzo (10 líneas por defecto) de uno o varios ficheros. De no especificarse al menos un fichero toman la entrada estándar.

Sintaxis:

```
tail [opciones] [ficheros]  
head [opciones] [ficheros]
```

Algunas opciones:

- f : para el caso de tail se ejecuta de forma sostenida, es decir continúa visualizando el final del fichero hasta que se interrumpe el proceso (Ctrl-c).

- q : no coloca los encabezamientos con el nombre de los ficheros cuando se indican varios (*quiet*).
- <n> : imprime las n últimas (primeras) líneas en lugar de las diez establecidas por defecto.

Ejemplos:

```
[root@fisct ~]# tail -f /var/log/messages
[root@fisct ~]# tail -20 /var/log/secure
[root@fisct ~]# head -15 /var/spool/mail/root
[root@fisct ~]# head -2 -q /etc/*.conf
```

6.4. Comando wc

El nombre del comando wc proviene de word count, sirve para contar palabras. Pero no sólo palabras, como veremos a continuación.

Sintaxis: wc [opción...] [archivo...]

Si se omite el argumento archivo, wc tomará los datos (naturalmente) de la entrada estándar.

La lista de opciones más importantes son las siguientes:

-c Contar bytes; **-l** Contar líneas; **-w** Contar palabras.

Como ejemplo, se pueden contar las líneas del archivo /etc/passwd y de esta manera se sabrá rápidamente cuantos usuarios tiene definido o creados en el sistema:

```
[root@fisct ~]# wc -l /etc/passwd
35 /etc/passwd
```

Tamaño que ocupa el archivo /etc/passwd

```
[root@fisct ~]# wc -l /etc/passwd
1639 /etc/passwd
```

6.5. Comando stat

El comando stat muestra las características de un fichero. Por ejemplo: su nombre, permisos, tamaño en *bytes*, número del *i-nodo* que lo representa, las fechas de modificación y acceso, el tipo, el dueño, el grupo, etc.

Sintaxis: stat [fichero...]

Ejemplos:

```
[root@fisct ~]# stat /boot/grub
File: "/boot/grub/"
Size: 4096          Blocks: 16          IO Block: 4096   directorio
Device: 301h/769d    Inode: 259330      Links: 2
Access: (0755/drwxr-xr-x)  Uid: (    0/    root)  Gid: (    0/    root)
Access: 2010-01-19 11:07:17.000000000 -0500
Modify: 2010-01-12 11:46:30.000000000 -0500
Change: 2010-01-12 11:46:30.000000000 -0500
```

```
[root@fisct ~]# stat /tmp/  
File: "/tmp/"  
Size: 1024      Filetype: Directory  
Mode: (1777/drwxrwxrwt)    Uid: (    0/  root)          Gid: (    0/  root)  
Device: 3,9           Inode: 4018                  Links: 5  
Access: 2010-01-19 08:49:33.000000000 -0500  
Modify: 2010-01-19 12:19:39.000000000 -0500  
Change: 2010-01-19 12:19:39.000000000 -0500
```

Lección 7

Empaquetar y comprimir Ficheros

7.1. Comando tar

El comando tar permite guardar o agrupar varios ficheros en un solo archivo y puede restablecer ficheros individuales a partir del archivo.

Sintaxis: tar [opción] [nombre.tar] [fichero(s)]

Entre las opciones del tar podemos mencionar:

-t, --list	lista todos los ficheros y directorios contenidos en un fichero tar
-x, --extract	extrae ficheros de un archivo
-c, --create	empaquetar un fichero tar
-d, --diff	encuentra las diferencias entre el archivo y el sistema de ficheros
-r, --append	añade ficheros al final de un archivo
-u, --update	sólo añade ficheros más recientes que la copia del archivo
-A, --catenate	añade ficheros tar a un archivo
--delete	borra de un archivo (no en cintas magnéticas!)
-f	indica el nombre asignado a un fichero tar
-v	verifica un fichero tar
-z	desempaquetar un fichero tar comprimido

Nota: Los ficheros empaquetados tienen como extensión **tar**.

Ejemplos:

Crear la siguiente estructura dentro del directorio /opt

```
[root@fisct ~]# cd /opt
```

```
computacion/
    hardware
    software
```

```
sistemas/
    computo01
    computo02
    computo03
```

Para empaquetar el directorio computacion como computacion.tar

```
[root@fisct ~]# tar -cvf computacion.tar computacion
```

Verificar:

```
[root@fisct ~]# tar -tf computacion.tar
```

Empaquetar los archivos computo01 y computo02

```
[root@fisct ~]# tar -cvf computo.tar sistemas/computo01 sistemas/computo02
```

Si desea agregar un nuevo archivo a computo.tar, por ejemplo computo03

```
[root@fisct ~]# tar -rvf computo.tar sistemas/computo03
```

7.2. Comando gzip

El comando gzip permite comprimir ficheros.

Sintaxis: gzip [opciones] fichero

Opciones:

- d descomprimir ficheros
- l lista todos los ficheros y muestra el % comprimido por cada fichero
- r opera recursivamente sobre directorios
- [0...9] grado de compresión

Ejemplo:

Comprimir el archivo computacion.tar ubicado dentro de /tmp/,

```
[root@fisct ~]# gzip -9 computacion.tar
```

Cabe indicar que un directorio no podrá ser comprimido directamente sin antes ser empaquetado.

7.3. Comando gunzip

Utilidad para descompresión de ficheros. Realiza la acción contraria que gzip. Descomprime archivos .gz devolviéndolos a su tamaño original. Equivale a ejecutar gzip -d 'fichero'.

Sintaxis: gunzip <fichero>

Ejemplo:

```
[root@fisct ~]# gunzip computacion.tar.gz
```

7.4. Comando bzip2

Utilidad de compresión de archivos más potente que gzip. El modo de funcionamiento es el mismo. bzip2 'fichero' para comprimir y bzip2 -d 'fichero' para descomprimir. Los ficheros comprimidos con este comando tendrán la extensión .bz2.

Sintaxis: bzip2 <fichero>

Ejemplo:

```
[root@fisct ~]# bzip2 computacion.tar
```

Para descomprimir utilizar el comando **bunzip2**

```
[root@fisct ~]# bunzip2 computacion.tar.bz2
```

Resumen

En esta unidad se ha descrito el manejo de los ficheros (archivos y directorios) permitiendo acceder o mostrar su contenido, así mismo, como prepararlos para empaquetarlos y comprimirlos para una mejor administración del espacio en el disco.

Lectura

El derecho a leer por Richard Stallman

Para Dan Halbert el camino a Tycho comenzó en la universidad, cuando Lissa Lenz le pidió prestado su ordenador. El de ella se había estropeado, y a menos que pudiese usar otro reprobaría su proyecto de fin de trimestre. No había nadie a quien se atreviera a pedírselo, excepto Dan.

Esto puso a Dan en un dilema. Tenía que ayudarle, pero si le prestaba su ordenador ella podría leer sus libros. Dejando de lado el riesgo de ir a la cárcel durante muchos años por dejar a otra persona leer sus libros, la simple idea le sorprendió al principio. Como a todo el mundo, se le había enseñado desde la escuela primaria que compartir libros era algo malo y desagradable, algo que sólo los piratas harían.

Además, no había muchas posibilidades de que la SPA (la "Software Protection Authority", o Autoridad de Protección del Software), no lo descubriese. En sus clases de programación Dan había aprendido que cada libro tenía un control de copyright que informaba de cuándo y dónde fue leído, y quién lo leía, a la oficina central de licencias (usaban esa información para descubrir piratas, pero también para vender perfiles personales a otras compañías). La próxima vez que su ordenador se conectase a la red, la oficina central de licencias lo descubriría. Él, como propietario del ordenador, recibiría el castigo más duro, por no tomar las medidas adecuadas para evitar el delito.

Lissa no necesariamente pretendía leer sus libros. Probablemente lo único que ella necesitaba era escribir su proyecto. Pero Dan sabía que ella provenía de una familia de clase media que a duras penas se podía permitir pagar la matrícula, sin pensar en las tasas de lectura. Leer sus libros podía ser la su única forma de terminar la carrera. Entendía la situación; él mismo había pedido un préstamo para pagar por los artículos de investigación que leía (el 10% de ese dinero iba a parar a los autores de los artículos, y como Dan pretendía hacer carrera en la universidad, esperaba que sus artículos de investigación, en caso de ser citados frecuentemente, le dieran los suficientes beneficios como para pagar el crédito).

Más tarde, Dan descubrió que hubo un tiempo en el que todo el mundo podía ir a una biblioteca y leer artículos, incluso libros, sin tener que pagar. Había investigadores que podían leer miles de páginas sin necesidad de becas de biblioteca. Pero desde los años 90 del siglo anterior, tanto las editoriales comerciales, como las no comerciales, habían empezado a cobrar por el acceso a los artículos. En el 2047, las bibliotecas de acceso público eran sólo un vago recuerdo.

Había formas de evitar los controles de la SPA y la oficina central de licencias, pero también eran ilegales. Dan había tenido un compañero de su clase de programación, Frank Martucci, que consiguió un depurador ilegal, y lo usaba para evitar el control de copyright de los libros. Pero se lo contó a demasiados amigos, y uno de ellos lo denunció a la SPA a cambio de una recompensa (era fácil tentar, para traicionar a sus amigos, a estudiantes con grandes deudas). En 2047 Frank estaba en la cárcel; pero no por pirateo, sino por tener un depurador.

Dan supo más tarde que hubo un tiempo en el que cualquiera podía tener un depurador. Incluso había depuradores libremente disponibles en la red. Pero los usuarios normales empezaron a usarlos para saltarse los controles de copyright, y finalmente un juez dictaminó que ese se había convertido en su uso práctico. Eso

quería decir que los depuradores eran ilegales y los programadores que los habían escrito fueron a parar a la cárcel.

Obviamente, los programadores necesitan depuradores, pero en el 2047 sólo había copias numeradas de los depuradores comerciales, y sólo disponibles para programadores oficialmente autorizados. El depurador que Dan había usado en sus clases de programación estaba detrás de un cortafuegos para que sólo se pudiese utilizar en los ejercicios de clase.

También se podía saltar el control de copyright instalando un núcleo del sistema modificado. Dan llegó a saber que hacia el cambio de siglo había habido núcleos libres, incluso sistemas operativos completos. Pero ahora no sólo eran ilegales, como los depuradores: no se podía instalar sin saber la clave de root del ordenador, cosa que ni el FBI ni el servicio técnico de Microsoft te darían.

Dan llegó a la conclusión de que simplemente no podía dejarle su ordenador a Lissa. Pero no podía negarse a ayudarle, porque estaba enamorado de ella. Cada oportunidad de hablar con ella era algo maravilloso. Y el hecho de que ella le hubiese pedido ayuda podría significar que sentía lo mismo por él.

Dan resolvió el dilema haciendo algo incluso más increíble, le dejó el ordenador, y le dijo su clave. De esta forma, si Lissa leía sus libros, la oficina central de licencias pensaría que quien estaba leyendo era él. Seguía siendo un delito, pero la SPA no lo detectaría automáticamente. Sólo podrían saberlo si Lissa lo denunciaba.

Si la universidad descubriese que le había dado su clave a Lissa significaría la expulsión para los dos, independientemente de para qué hubiese usado ella la clave. La política de la universidad era que cualquier interferencia con sus métodos de control sobre el uso de los ordenadores era motivo para una acción disciplinaria. No importaba si se hubiera hecho o no algún daño, el delito era el hecho de dificultar el control. Se asumía que esto significaba que se estaba haciendo algo prohibido, y no necesitaban saber qué.

En general los estudiantes no eran expulsados por eso -no directamente-. En su lugar se les prohibía el acceso a los ordenadores de la universidad, lo que inevitablemente significaría reprobando todas sus asignaturas.

Dan supo más tarde que ese tipo de políticas en la universidad empezaron en la década de 1980, cuando los estudiantes comenzaron a usar ordenadores masivamente. Antes de eso, las universidades tenían una actitud diferente: sólo se penalizaban las actividades dañinas, no las que eran meramente sospechosas.

Lissa no denunció a Dan a la SPA. Su decisión de ayudarle llevó a que se casasen, y también a que cuestionasen lo que les habían enseñado cuando eran niños sobre el pirateo. Empezaron a leer sobre la historia del copyright, sobre la Unión Soviética y sus restricciones sobre las copias, e incluso sobre la constitución original de los Estados Unidos. Se mudaron a Luna, donde se encontraron con otros que de la misma forma intentaban librarse del largo brazo de la SPA. Cuando empezó el Levantamiento de Tycho en 2062, el derecho universal a leer se convirtió en uno de sus objetivos fundamentales.

Nota del autor

El derecho a leer es una batalla que se está librando hoy en día. Nuestra forma de vida actual podría tardar 50 años en desvanecerse, pero muchas de las leyes y

prácticas descritas más arriba ya han sido propuestas, o por la administración Clinton o por las editoriales.

Hasta hace poco había una excepción: la idea de que el FBI y Microsoft se guardaran las claves de root de los ordenadores personales, y no dejaran obtenerlas a los usuarios no fue propuesta hasta 2002. A esto se le llamó "computación confiable" o "palladium".

En 2001, el senador Hollings, apoyado financieramente por la Disney, propuso un proyecto de ley, llamado SSSCA, que requeriría que cada ordenador nuevo tuviera restricciones para efectuar copias, que los usuarios no podrían evitar. En la misma línea que la del chip Clipper y otras propuestas similares del gobierno de los EE.UU. sobre custodia de claves de encriptación, esta es una tendencia a largo plazo: los sistemas de ordenadores se configuran cada vez más para dar control sobre el ordenador a terceras partes en lugar de a las personas que realmente lo utilizan. La SSSCA ha sido llamada desde entonces la CBDTPA (denotando "Consume But Don't Try Programming Act", "Consumir Pero Ni Intentes Programar").

En 2001 los EE.UU. comenzaron a intentar el uso del propuesto Tratado del Área de Libre Comercio de las Américas (ALCA) para imponer las mismas reglas en todos los países del hemisferio occidental. El ALCA es uno de los tratados llamados "de libre comercio" realmente diseñados para darles a las empresas mayor poder frente a los gobiernos democráticos; imponer leyes tales como la DCMA es típico de este espíritu. La Electronic Frontier Foundation le solicita a las personas que expliquen a esos gobiernos por qué deberían oponerse a tales planes.

La SPA, que realmente significa "Software Publisher's Association" (Asociación de Editores de Software), ha sido reemplazada en este rol policial por la BSA, o "Business Software Alliance". Esta no es una fuerza policial, pero extraoficialmente actúa como si lo fuera. Utilizando métodos que recuerdan a la antigua Unión Soviética, invita a la gente a informar a sus compañeros de trabajo y amigos. En 2001 una campaña de terror de la BSA en Argentina realizó amenazas veladas de que aquellos que compartieran programas de ordenador terminarían siendo violados en prisión.

Cuando se escribió esta historia, La SPA estaba amenazando a pequeños proveedores de Internet (ISP) para que les permita controlar a sus usuarios. La mayoría de ellos cedieron al ser amenazados, ya que no podían costearse la pelea judicial en los tribunales (Atlanta Journal-Constitution, 1 Oct 96, D3). Al menos un ISP, "Community ConneXion" en Oakland CA, se negó a aceptar las presiones, y fue eventualmente demandado. La SPA luego retiró la demanda, pero obtuvo la DMCA ("Digital Millennium Copyright Act", o Ley del Copyright del Milenio Digital), la cual les dio el tipo de poder que buscaban.

Las políticas de seguridad descritas arriba no son imaginarias. Por ejemplo, un ordenador de una universidad del área de Chicago muestra el siguiente mensaje al conectarse al sistema (las comillas están en el original):

"Este sistema sólo puede ser utilizado por usuarios autorizados. Cualquier individuo que use esta sistema sin autorización, o excediendo su autorización está sujeto a ser monitorizado por el personal del sistema. Al controlar usuarios realizando actividades no autorizadas o durante el mantenimiento del sistema, las actividades de usuarios autorizados pueden ser monitorizadas. Cualquiera que use este sistema acepta expresamente tal monitorización y queda advertido de que si ese control revela posibles indicios de actividades ilegales o violación de las normas de la Universidad, el personal de mantenimiento del sistema puede proporcionar esas evidencias a las autoridades de la Universidad o a las fuerzas de seguridad".

Esta es una aproximación interesante a la Cuarta Enmienda: forzar a los usuarios a declinar por adelantado los derechos en ella contemplados.

Fuente [1].

Autoevaluación

1. Marcar la respuesta correcta:

1.1. No corresponde a un comando para mostrar el contenido de un archivo:

- a. ls ()
- b. more ()
- c. grep ()
- d. zmore ()
- e. cat ()
- f. less ()

1.2. Permite realizar la búsqueda de ficheros:

- a. slocate ()
- b. updatedb ()
- c. ls ()
- d. find ()
- e. LOCATE ()
- f. "d" y "e" ()

1.3. Para desempaquetar y descomprimir el fichero.tar.zip debe emplear:

- a. tar -xjvf fichero.tar.zip ()
- b. gunzip -xzvf fichero.tar.zip ()
- c. tar -xzvf fichero.tar.zip ()
- d. bunzip2 -xjvf fichero.tar.zip ()
- e. tar -xvf fichero.tar.zip ()
- f. unzip fichero.tar.zip ()

1.4. Para desempaquetar y descomprimir fichero.tar.Z" debe emplear:

- a. tar -xjvf fichero.tar.Z ()
- b. gunzip -xzvf fichero.tar.Z ()
- c. tar -xzvf fichero.tar.Z ()
- d. bunzip2 -xjvf fichero.tar.Z ()
- e. tar -xvf fichero.tar.Z ()
- f. unzip fichero.tar.Z ()

1.5. Permite empaquetar y comprimir ficheros:

- a. bzip2 ()
- b. tar ()
- c. unzip ()
- d. gzip ()
- e. bunzip2 ()
- f. zip ()

1.6. Para añadir información a un archivo tar. ¿Qué deber hacer?

- a. Usar el comando tar con la opción -a ()
- b. Usar el comando append ()
- c. Usar el comando add ()
- d. Usar el comando tar con la opción -t ()
- e. Usar el comando tar con la opción -r ()
- f. N.A. ()

1.7. Para extraer el contenido del fichero php-5.2.5.tar.bz2 debe emplear:

- a. gunzip -xzvf php-5.2.5.tar.bz2 ()
- b. tar -xzvf php-5.2.5.tar.bz2 ()
- c. tar -xvf php-5.2.5.tar.bz2 ()
- d. bunzip2 -xjvf php-5.2.5.tar.bz2 ()
- e. tar -xvf php-5.2.5.tar.bz2 ()
- f. tar -xjvf php-5.2.5.tar.bz2 ()

2. RELACIONE E IDENTIFIQUE LA RESPUESTA CORRECTA. EN LOS CASILLEROS

- a. Permite agrupar los contenidos de dos o más archivos en uno solo () locate -u
- b. Es el encargado de ser el intérprete de comandos utilizado en el sistema () find / -name fichero
- c. Equivalente del comando updatedb () finger
- d. Nos muestra información del total de sesiones abiertas en el ordenador () cat file1 file2 >> fileextenso
- e. Gestor de arranque que permite arranque dual de dos o más sistemas operativos instalados en un único ordenador () terminal
- f. Permite buscar uno o más ficheros relacionado con un determinado patrón () bash
() locate fichero
() slocate
() grub
() who
() grep gnome file1 file | more

Solucionario

1. Marcar la respuesta correcta

- 1.1. a
- 1.2. d
- 1.3. c
- 1.4. c
- 1.5. b
- 1.6. e
- 1.7. f

2. RELACIONE E IDENTIFIQUE LA RESPUESTA CORRECTA. EN LOS CASILLEROS

- a. Permite agrupar los contenidos de dos o más archivos en uno solo (c) locate -u
- b. Es el encargado de ser el intérprete de comandos utilizado en el sistema (f) find / -name fichero
- c. Muestra información de una usuario () stat
- d. Nos muestra información del total de sesiones abiertas en el ordenador (a) cat file1 file2 >> fileextenso
- e. Gestor de arranque que permite arranque dual de dos o más sistemas operativos instalados en un único ordenador () terminal
- f. Permite buscar uno o más ficheros relacionado con un determinado patrón (b) bash
 - (c) finger
 - () slocate
 - (e) grub
 - (d) who
 - () grep gnome file1 file | more

Bibliografía

- [1]. Stallman, Richard (1997) "B The right to read", Revista Communications of the ACM, Volumen 40, Número 2, Pp. 85-87. ISSN: 0001-0782.

Enlaces

- Baig Viñas, Roger y Aulí Llinás (2003) "Sistema Operativo GNU/Linux Básico" *Formación de Posgrado de la UOC - Máster oficial de Software libre*.
http://www.uoc.edu/masters/oficiales/master_oficial_software_libre/master_oficial_software_libre_materiales.htm
- Kirch, Olaf y Dawson, Ferry (2002) "Guía de Administración de Redes con Linux". *O'Reilly (printed version) (c) 2000 O'Reilly & Associates. Proyecto LuCAS por la traducción al español*.
<http://es.tldp.org/Manuales-LuCAS/GARL2/garl2/>
- Red Hat, Inc (2005) "Red Hat Enterprise Linux 4 - Introducción a la administración de sistemas"
<http://www.redhat.com/docs/manuals/enterprise/RHEL-4-Manual/es/admin-guide/>
- Red Hat, Inc (2005) "Red Hat Enterprise Linux 4 - Manual de Referencia"
<http://www.redhat.com/docs/manuals/enterprise/RHEL-4-Manual/es/ref-guide/>

UNIDAD III

USUARIOS y GRUPOS

La unidad tiene como propósito que el estudiante conozca cómo crear, modificar y eliminar cuentas de usuarios y grupo en los sistemas GNU/Linux, valorando la importancia de los conocimientos para su desarrollo académico. Contiene:

- Usuarios y Grupos

Lección 8

Usuarios y grupos

8.1. Conceptos básicos

La administración del sistema GNU/Linux dependerá del control de los usuarios y grupos siendo elementos claves para su funcionamiento [1].

Los usuarios son cuentas que guardan relación a un usuario físico en particular o cuentas que existen para ser usadas por aplicaciones específicas (exim, apache, mysql, entre otros) [3].

Los grupos son expresiones lógicas en la organización, agrupando a usuarios para un propósito común. Los usuarios dentro de un mismo grupo pueden leer, escribir o ejecutar archivos que pertenecen al mismo grupo [2].

Cada usuario y grupo tiene un número de identificación único llamado userid (UID) y un groupid (GID) respectivamente [1].

Al crearse un fichero se le asigna un usuario y un grupo. De la misma forma se asignan los permisos de lectura, escritura y ejecución para el propietario del archivo, para el grupo y para cualquier otro usuario en un host. El usuario y el grupo de un fichero particular, así como los permisos en ese fichero, pueden ser cambiados por el usuario root o por el creador del fichero [1] [2].

Las cuentas de usuarios creados se registran en el fichero passwd que se ubica en el directorio /etc

```
[root@server ~]# cd /etc  
[root@server ~]# more passwd  
root:x:0:0:root:/root:/bin/bash  
bin:x:1:1:bin:/bin:  
adm:x:3:4:adm:/var/adm:  
lp:x:4:7:lp:/var/spool/lpd:  
mail:x:8:12:mail:/var/spool/mail:
```

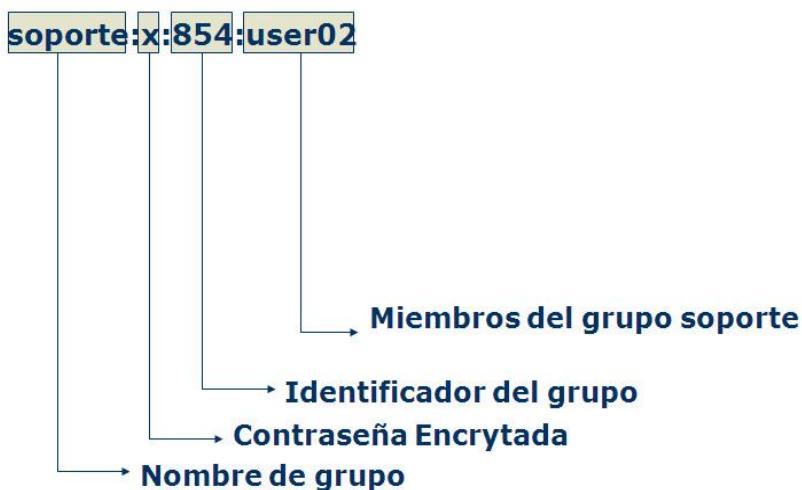
Cada línea representa a una cuenta de usuario conformado por los siguientes elementos:



Los grupos de trabajo se registran en el fichero group que se ubica en el directorio /etc

```
[root@server ~]# cd /etc
[root@server ~]# more group
root:x:0:root
bin:x:1:root,bin,daemon
sys:x:3:root,bin,adm
adm:x:4:root,adm,daemon
ftp:x:50:
nobody:x:99:
users:x:100:
soporte:x:854:
```

Cada línea representa a un grupo conformado por los siguientes elementos:



8.2. Creación de Cuentas de Usuarios y Grupos

Para crear cuentas de usuarios debe emplear el comando **useradd**.

Sintaxis:

- useradd username
- useradd [opciones] username

Opciones:

- c: Se utiliza para agregar el nombre completo o algún comentario referente al usuario.
- d: Crea el home particular del usuario. Si no se especifica se creará automáticamente con el mismo nombre de la cuenta de usuario dentro de la carpeta /home.
- s: Asignación del shell, con esta opción habilita o inhabilita el acceso al sistema a una cuenta de usuario.
- g: Asignar el grupo principal, cada cuenta de usuario se le asignará un grupo principal, en caso de no emplearse esta opción se creará un grupo con el mismo nombre del username y éste será asignado al usuario.
- G: Agregar otros puede asignarles otros grupos a la cuenta de usuario. Son los llamados grupos secundarios.
- u: Permite definir el UID (Identificador de Usuario).
- M: No crea el directorio principal.

username (login): es el nombre de la cuenta de usuario que será utilizado para el ingreso al sistema.

Nota.- Sólo el usuario root tiene el permiso de crear usuarios y grupos.

Para crear grupos debe emplear el comando: **groupadd**.

Sintaxis:

- groupadd groupname
- groupadd [opciones] groupname

Opciones:

- g: Identificador de grupo (GID), el cual debe ser único y mayor que 499.
- r: Crea un grupo de sistema con un GID menor que 500.

groupname: es el nombre de grupo a crearse.

Ejemplos:

a. Crear la cuenta soporte

```
[root@server ~]# useradd soporte
```

Debe ingresar el password para el usuario soporte y luego confirmarla.

```
[root@server ~]# passwd soporte
Changing password for user soporte
New UNIX password:
Retype new UNIX password:
passwd: all authentication tokens updated successfully.
```

Nótese que se ha creado la cuenta donde no se ha utilizado ninguna de las opciones. Para realizar una verificación de la cuenta emplearemos el comando finger.

```
[root@server ~]# finger soporte
Login: soporte                                Name: (null)
Directory: /home/soporte                          Shell: /bin/bash
On since lun abr 23 18:24 (PET) on pts/0 from 192.168.0.7
No mail.
```

El resultado nos muestra:

- Login o username del usuario,
- Name: se encuentra vacío dado que no se ingresó un nombre descriptivo para la cuenta,
- Directory: es el directorio creado para la cuenta, en este caso coincide con el login,
- Shell: Por defecto se le asigna el shell bash para las cuentas recién creadas. Este shell se encuentra dentro de la carpeta /bin y donde además se encuentran otros tipos de shell como sh, csh, etc.,
- La tercera línea muestra si esta cuenta ha sido utilizada, desde qué lugar y la fecha que ingresó. En este caso esta cuenta fue utilizada la última vez el 23 de abril a las 18:24 horas y se conectó desde el ordenador con IP 192.168.0.7

- La cuarta línea se usa cuando hay un servidor de correo instalado y configurado. Muestra si el usuario realiza mantenimiento de su buzón de correo.

Abrir un nuevo terminal (presiónela la combinación de teclas CTRL+ALT+F2) para comprobar el ingreso de la cuenta.

Para regresar a la interfaz gráfica, presionar la combinación de teclas CTRL+ALT+F7.

b. Crear la cuenta para el usuario Juan Ramón Rojas

```
[root@server ~]# useradd -c "Juan Ramon Rojas" jramonr  
[root@server ~]# passwd jramonr  
Changing password for user jramonr  
New UNIX password:  
Retype new UNIX password:  
passwd: all authentication tokens updated successfully.
```

Se ha agregado el nombre completo del usuario utilizando la opción -c, el username está representado por el primer carácter del nombre más el apellido paterno y el primer carácter del apellido materno. Usted puede definir cómo crear el username. Se recomienda seguir con un estándar que se aplique para la creación de nuevas cuentas.

c. Utilice el comando finger y complete

```
[root@server ~]# finger jramonr  
Login: _____ Name: _____  
Directory: _____ Shell: _____  
_____
```

d. Crear la cuenta para el usuario Jacqueline Domínguez Valverde, donde su directorio privado debe ser creado con el nombre más el apellido paterno y el username primer carácter del nombre seguido del apellido paterno y el primer carácter del apellido materno

```
[root@server ~]# useradd -d /home/jaquelinedominguez -c "Jacqueline  
Domínguez Valverde" jdominguezv  
[root@server ~]# passwd jdominguezv  
Changing password for user jdominguezv  
New UNIX password:  
Retype new UNIX password:  
passwd: all authentication tokens updated successfully.
```

e. Se necesita instalar el programa spamassassin, y uno de los requisitos es crear la cuenta spam, para poder ser instalado. Esta cuenta no tendrá acceso para ingresar al sistema.

Puede Ud. crear la cuenta utilizando cualquiera de las líneas de comando:

```
[root@server ~]# useradd -s /sbin/nologin spam  
[root@server ~]# useradd -c "Programa Spamassassin" -s /sbin/nologin spam  
[root@server ~]# useradd -s /bin/false spam  
[root@server ~]# useradd -c "Programa Spamassassin" -s /bin/false spam
```

```
[root@server ~]# useradd spam -s /sbin/nologin
```

```
[root@server ~]# useradd spam -s /bin/false
```

Nótese que como shell asignado es un /sbin/nologin o /bin/false donde no permiten que el usuario ingrese al sistema. Dado que no cumplen la función de un shell convencional como el bash. Para este tipo de cuenta no es necesario ingresar un password.

f. Al administrador de la red de la empresa CAGG S.A le han encargado crear una cuenta para Sonia Candela Malpica, nuevo integrante del área de marketing. De acuerdo con la asignación del identificador de usuario le correspondería el 1050.

```
[root@server ~]# useradd -c "Sonia Candela Malpica" -u 1050 scandela
```

```
[root@server ~]# passwd scandela
```

Changing password for user scandela

New UNIX password:

Retype new UNIX password:

```
passwd: all authentication tokens updated successfully.
```

Para la creación de la cuenta, una de las opciones utilizadas es -u, para asignar el identificador de usuario 1050

g. Hay un nuevo Gerente de TI en la empresa TECHNOLOGY SAV, es el Ing. José Edmundo Dediós Castillo, se necesita crearle su cuenta de usuario y su grupo principal es TISYSTEM donde están los demás gerentes que pueden compartir información.

Primero, asegúrese si el grupo tisystem existe. Se empleará el comando groupadd para la creación de grupos.

```
[root@server ~]# groupadd tisystem
```

Si el grupo no esta creado, no debe mostrar ningún mensaje de alerta. Caso contrario debe mostrar: "group tisystem exists", donde le indica la existencia del grupo tisystem

```
[root@server ~]# groupadd tisystem
```

```
groupadd: group tisystem exists
```

Ahora proceda a crear a la cuenta para José Edmundo Dediós Castillo

```
[root@server ~]# useradd -c "José Edmundo Dediós Castillo" -g tisystem josededios
```

```
[root@server ~]# passwd josededios
```

Changing password for user josededios

New UNIX password:

Retype new UNIX password:

```
passwd: all authentication tokens updated successfully.
```

h. Utilizando el comando finger, responde lo siguiente:

- Shell por defecto asignado a la cuenta josededios _____
- ¿Donde se localiza el home directory o directorio del usuario josededios y con qué nombre? _____
- ¿Qué opción debió utilizar si quería asignar o definir el nombre del home directory? _____

- ¿Qué sucede sino definía el nombre o la descripción del usuario? _____
- El sistema permite crear el grupo ITSYSTEM (todo en mayúscula) _____
- Es necesario definir un shell para que el usuario josededios pueda ingresar al sistema o ¿qué necesita para tenerlo habilitado? _____
- Equivale utilizar la opción -c y -C ¿porqué? _____

i. Hay una cuenta que debe crearse para Zheila Torres Zamudio, donde su grupo principal es desarrollo y además estará dentro de otros grupos como sistemas y soporte.

Debe crear los grupos desarrollo, sistemas y soporte

```
[root@server ~]# groupadd desarrollo
[root@server ~]# groupadd sistemas
[root@server ~]# groupadd soporte
```

Proceda a crear la cuenta:

```
[root@server ~]# useradd -g desarrollo -G sistemas,soporte -c "Zheila Torres
Zamudio" ztorresz
[root@server ~]# passwd ztorresz
Changing password for user ztorresz
New UNIX password:
Retype new UNIX password:
passwd: all authentication tokens updated successfully.
```

Nótese que para asignarle más de un grupo secundario la separación se realiza a través de coma (,) y no debe dejar espacio en blanco entre cada grupo secundario.

j. Crear la cuenta exim para el programa del servidor de correos llamado exim, esta cuenta no debe tener un home directory y además no tiene acceso al sistema.

De acuerdo con lo indicado, esta cuenta va a existir, pero con restricciones.

```
[root@server ~]# useradd -s /bin/false -M exim
```

Al emplear la opción -M estamos indicando que no cree ningún directorio para la cuenta exim. Compruebe haciendo un ls -l al directorio /home.

```
[root@server ~]# ls -l /home
```

Sin embargo al hacer un finger a la cuenta exim, vemos lo siguiente:

```
[root@server ~]# finger exim
Login: exim                                Name: (null)
Directory: /home/exim                         Shell: /bin/false
No Plan
No Mail
```

En Directory indica /home/exim, esto no indica que el home directory esté creado, sino que este directorio debiera estar creado en /home.

La cuenta no requiere del ingreso de un password.

k. Crear el grupo analistas cuyo identificador es 1000

```
[root@server ~]# groupadd -g 1000 analistas
```

I. Crear la siguiente cuenta de usuario con los siguientes datos:

Descripción	:	Sofía Corrales Ronseros
Directorio Personal	:	sofiacorrales
Grupo Primario	:	gerente
Grupo Secundario	:	director y consultor
Identificador de Usuario	:	a partir del 2001
Cuenta de Usuario	:	scorrales

Defina todos los pasos que debe realizar.

8.3. Modificar cuentas de Usuarios y Grupos

Para modificar cuentas de usuario, debe emplear el comando: **usermod**. **Sólo el usuario root tiene el permiso para modificar las cuentas de usuario y grupo.**

Sintaxis: usermod [opciones] username

Opciones:

- c: Se utiliza para agregar o modificar el nombre completo o algún comentario referente al usuario.
- d: Modificar el home particular del usuario. Esta va acompañado al final de la opción -m.
- s: Modificar el Shell.
- g: Modificar el grupo principal.
- G: Agregar o modificar grupos secundarios. Para asignar mas de un grupos esta va separado por coma (,) sin dejar ningún espacio en blanco.
- u: Modificar el UID (Identificador de Usuario).
- l: Modificar el login del usuario.
- L: Bloquear el password del usuario. Coloca el símbolo “!” al inicio de la clave encriptada que se encuentra en el archivo /etc/shadow.
- U: Desbloquear el password del usuario. Retira el símbolo “!” al inicio de la clave encriptada que se encuentra en el archivo /etc/shadow.

Cuando se actualiza algunos de los datos, no es necesario modificar el password de la cuenta del usuario dado que ésta sigue siendo igual.

Para modificar grupos, debe emplear el comando: **groupmod**

Sintaxis: groupmod [opciones] groupname

Opciones:

- g: Identificador de grupo (GID), el cual debe ser único y mayor que 499.
- n: Cambiar el nombre de grupo.

groupname: es el nombre de grupo a crearse.

Ejemplos:

a. Modificar la cuenta soporte donde se indique “Área de Soporte”.

```
[root@server ~]# usermod -c "Área de Soporte" soporte
```

Para realizar una verificación de la cuenta emplearemos el comando finger

```
[root@server ~]# finger soporte
Login: soporte                                Name: Área de Soporte
Directory: /home/soporte                         Shell: /bin/bash
On since lun abr 23 18:24 (PET) on pts/0 from 192.168.0.7
No mail.
```

El resultado nos muestra que se ha actualizado el campo Name.

b. Se necesita modificar el apellido paterno de Juan Ramón Rojas, donde sus verdaderos datos son Juan Riquelme Rojas, además esto afectaría a su home y su login.

En este caso hay que modificar sus datos, su home y su login

```
[root@server ~]# usermod -c "Juan Riquelme Rojas" -d /home/jriquelmer -m -l jriquelme jramonr
```

Se ha utilizado las opciones -c, -d (fíjese que se agregó al final la opción -m) y -l

c. Utilice el comando finger y complete los datos que muestra.

```
[root@server ~]# finger jramonr
Login: _____                                Name: _____
Directory: _____                            Shell: _____
_____
```

d. La señorita Jacqueline Domínguez Valverde saldrá de vacaciones por un mes y ha pedido que su cuenta sea bloqueada para el ingreso al sistema.

Antes de efectuar el cambio ingrese la siguiente línea de comando:

```
[root@server ~]# grep jdominguezv /etc/shadow
jdominguezv:$1$GVIhjT8h$NBIH9y5AfAbl/zpbRsAuV/:13630:0:99999:7:::
```

Debe mostrarle algo similar con respecto a la clave encriptada del usuario.

Abrir un terminal e ingrese con la cuenta, al igual que su password (si no recuerda el password, como usuario root modifíquelo: passwd jdominguezv). Una vez que comprobó el ingreso proceda a cerrar la sesión con el comando exit.

```
[jdominguezv@server ~]$ exit
```

Regrese a la consola o entorno donde tenga una sesión abierta como usuario root y escriba la siguiente línea de comando:

```
[root@server ~]# useradd -L jdominguezv
```

Repita la línea de comando:

```
[root@server ~]# grep jdominguezv /etc/shadow  
jdominguezv:!$1$GVIhjT8h$NBIH9y5AfAbl/zpbRsAuV/:13630:0:99999:7:::
```

Nótese que se ha agregado el símbolo “!” al inicio de la clave.

Cambie de terminal y trate de ingresar.

Regrese a la consola o entorno donde tenga una sesión abierta como usuario root.

Para habilitarlo la cuenta ingrese la siguiente línea de comando:

```
[root@server ~]# useradd -U jdominguezv
```

Repita la línea de comando:

```
[root@server ~]# grep jdominguezv /etc/shadow  
jdominguezv:$1$GVIhjT8h$NBIH9y5AfAbl/zpbRsAuV/:13630:0:99999:7:::
```

Nótese que se ha eliminado o retirado el símbolo “!” al inicio de la clave.

Cambie de terminal y compruebe si ingresa (la clave de la cuenta sigue siendo la misma). Una vez que comprobó el ingreso proceda a cerrar la sesión con el comando exit.

```
[jdominguezv@server ~]$ exit
```

Regrese a la consola o entorno donde tenga una sesión abierta como usuario root.

Otra forma es cambiando el shell del usuario, por un /sbin/nologin o /bin/false. Ingrese la siguiente línea de comando:

```
[root@server ~]# usermod -s /bin/false jdominguezv  
o  
[root@server ~]# usermod -s /sbin/nologin jdominguezv
```

Cambie de terminal y trate de ingresar.

Regrese a la consola o entorno donde tenga una sesión abierta como usuario root.

Con el comando passwd también puede bloquear la cuenta. Cambie primero el shell del usuario jdominguezv

```
[root@server ~]# usermod -s /bin/bash jdominguezv
```

Cambie de terminal y compruebe si ingresa (la clave de esta cuenta sigue siendo la misma). Regrese a la consola o entorno donde tenga una sesión abierta como usuario root.

Ahora, utilizando el comando passwd ingrese lo siguiente:

```
[root@server ~]# passwd -l jdominguezv
```

Ingresé la siguiente línea a continuación:

```
[root@server ~]# grep jdominguezv /etc/shadow  
jdominguezv:!!$1$GVIhjT8h$NBIH9y5AfAbl/zpbRsAuV/:13630:0:99999:7:::
```

Nótese que se ha agregado el símbolo “!!” al inicio de la clave.

Cambie de terminal y trate de ingresar. Regrese a la consola o entorno donde tenga una sesión abierta como usuario root.

Para habilitar la cuenta escriba la siguiente línea de comando:

```
[root@server ~]# passwd -u jdominguezv
```

Repita la línea de comando:

```
[root@server ~]# grep jdominguezv /etc/shadow  
jdominguezv:$1$GVIhjT8h$NBIH9y5AfAbl/zpbRsAuV/:13630:0:99999:7:::
```

Nótese que se ha eliminado o retirado el símbolo “!!” al inicio de la clave.

Cambie de terminal y compruebe si ingresa (la clave de esta cuenta sigue siendo la misma).

Regrese a la consola o entorno donde tenga una sesión abierta como usuario root.

e. Se necesita habilitar la cuenta del programa spamassassin, (usuario spam) para realizar modificaciones en el programa y solo este usuario lo puede realizar.

Indique que línea de comando debe utilizar

```
[root@server ~]# _____
```

f. El administrador de la red de la empresa CAGG S.A. ingreso para el usuario Sonia Candela Malpica, un UID equivocado, el UIGV el 1100 el cual le corresponde.

```
[root@server ~]# usermod -u 1100 scandela
```

g. Al Gerente de TI en la empresa TECHNOLOGY SAV, es el Ing. José Edmundo Dediós Castillo, modifique su grupo principal a TICSYSTEM y agregarlo a los grupos sistemas y desarrollo.

Asegúrese de que los grupos ticsystem, sistemas y desarrollo existen. Si no emplee el comando groupadd para la creación de los grupos.

```
[root@server ~]# groupadd tisystem  
[root@server ~]# groupadd sistemas  
[root@server ~]# groupadd desarrollo
```

Ahora, proceda a modificar la cuenta.

```
[root@server ~]# usermod -g ticsystem -G sistemas,desarrollo josededios
```

h. Del usuario Zheila Torres Zamudio cambie su grupo principal a proyecto y para otros grupos solo debe estar en desarrollo.

Defina la(s) línea(s) de comando:

```
[root@server ~]# _____  
_____
```

i. Defina los pasos para habilitar el ingreso de la cuenta exim.

[root@server ~]# _____

8.4. Eliminar cuentas de Usuarios y Grupos

Para eliminar cuentas de usuario debe emplear el comando: userdel. Sólo el usuario root tiene el permiso para eliminar cuentas de usuario y grupo.

Sintaxis: userdel [opcion] username

Opciones:

-r: Se utiliza eliminar el home de los usuarios creados en el sistema

Nota.- Una cuenta de usuario para ser eliminada no tiene que estar activo.

Para eliminar grupos debe emplear el comando: **groupdel**.

Sintaxis: groupdel groupname

Nota.- Un grupo para ser eliminado no deberá ser grupo principal de una cuenta de usuario.

Ejemplos:

[root@server ~]# userdel soporte # Elimina solo la cuenta de usuario

[root@server ~]# userdel -r jriquelmente # Elimina la cuenta de usuario y su
home

[root@server ~]# groupdel desarrollo

Resumen

En esta unidad, se ha descrito los pasos esenciales para la creación, modificación y eliminación de cuentas de usuarios y grupos. Son elementos esenciales para el funcionamiento del sistemas GNU/Linux, la mayoría de aplicaciones requieren de una cuenta de usuario. Así mismo, se ha realizado una revisión a los archivos principales de su configuración.

Lectura

Mitos y realidades: Linux y los Virus

El debate sobre Linux y los virus no es algo nuevo. Cada cierto tiempo vemos un correo en una lista preguntando si existen virus para Linux; y automáticamente alguien responde afirmativamente y alega que si no son mas populares es porque Linux no está tan extendido como Windows. También son frecuentes las notas de prensa de desarrolladores de antivirus diciendo que sacan versiones contra los virus de Linux.

Mi experiencia como administrador:

En más de diez años que llevo administrando Linux, con instalaciones en cientos de máquinas de centro de cálculo, laboratorio de alumnos, empresas, etc.

- Nunca me ha “entrado” un virus.
- Nunca he conocido a alguien que le haya ocurrido
- Nunca he conocido a alguien que haya conocido a alguien que le hay ocurrido

Conozco a más gente que ha visto al monstruo del Lago Ness a que haya visto virus para Linux.

Personalmente, reconozco que he sido un temerario, y he lanzado varios programas que los autoprogramados “especialistas” denominan “virus para Linux” -en adelante, los denominaré virus, para no hacer pedante el texto-, desde mi cuenta habitual contra mi máquina, para ver si es posible un virus: tanto el virus bash que circula por ahí -y que, por cierto, no me infectó ningún fichero-, como un virus que se hizo muy famoso, y salió en la prensa. Intenté instalarmelo; y después de veinte minutos de trabajo, me rendí cuando vi que entre sus exigencias estaba tener el directorio tmp en una partición del tipo MSDOS. Personalmente, no conozco a nadie que cree una partición específica para tmp y la formatee en FAT.

De hecho, algunos supuestos virus que he probado para Linux necesitan un nivel de conocimientos altos y la clave de root para ser instalados.

Podríamos calificar, cuanto menos, de “cutre” un virus si necesita nuestra intervención activa para que nos infecte la máquina. Además, en algún caso requieren amplios conocimientos de UNIX y la clave de root; lo que está bastante lejos de la instalación automática que se le supone.

Fuente [4].

Autoevaluación

1. Del siguiente listado, marcar cuál permite tener ingreso al sistema :

- | | |
|--|-----|
| a. operator:x:11:0:operator:/root:/sbin/nologin | [] |
| b. mysql:x:501:501::/home/mysql:/bin/false | [] |
| c. a0223657:x:1015:10:Melo Perez Brigitte:/home/a0223657:/bin/sh | [] |
| d. d8552196:x:5543:10:Torres Rios Alinson:/home/d8552196:/bin/bash | [] |
| e. Irojas:x:7185:10::/home/personal/Irojas:/bin/false | [] |
| f. r_soto:x:7480:10:Raul Soto Garibay:/home/r_soto:/bin/bash | [] |

2. Por encargo de la Oficina de Personal, se pide que habilite una cuenta de usuario para el nuevo personal que acaba de ingresar. Todos ellos ya fueron notificados al área que laborarán.

GRUPOS: contadores, asesores

USUARIO	NOMBRE COMPLETO	ÁREA DE TRABAJO	OTRO GRUPO
svasques	Sonia Vasquez Villar	contadores	asesores
agonzales	Adrian Gonzales Guisado	contadores	
ecardenas	Erika Cárdenas Cuba	asesores	

* Todos los usuarios tienen shell bash

* Asignar la clave correspondiente para cada usuario

Responder:

- Línea de comando para crear los grupos
- Línea de comando para crear el usuario Sonia Vasquez Villar
- Línea de comando para crear el usuario Adrian Gonzales Guisado
- Línea de comando para crear el usuario Erika Cárdenas Cuba pero con su directorio en /home/asesores/ecardenas y además estará temporalmente deshabilitado la cuenta.
- ¿Cómo determino el total de usuarios creados en el sistema? Escriba la línea de comando

3. Se necesita crear la siguiente cuenta de usuario con los siguientes datos:

Cuenta de Usuario	:	scorrales
Descripción	:	Sofia Corrales Ronseros
Grupo Primario	:	gerente
Grupo Secundario	:	director

Marque la opción u opciones que permitan crear dicha cuenta:

- groupadd gerente; groupadd director; useradd -c "Sofia Corrales Ronseros" -g gerente -G director -l /bin/false scorrales
- groupadd gerente director; useradd -c "Sofia Corrales Ronseros" -g gerente -G director scorrales
- groupadd gerente; useradd -c "Sofia Corrales Ronseros" -d /home/scorrales -g gerente -G director -s /bin/bash scorrales
- groupadd gerente; groupadd director; useradd -c "Sofia Corrales Ronseros" -G gerente -g director scorrales
- groupadd gerente; groupadd director; useradd -c "Sofia Corrales Ronseros" -g gerente -G director scorrales
- groupadd gerente; groupadd director; useradd -c "Sofia Corrales Ronseros" -G gerente -g director -s /bin/false scorrales

4. Se necesita modificar la cuenta de Sonia Cuba Maldonado (VER MODIFICAR POR) el cual muestra los siguientes datos (VER DATOS ACTUALES):

	DATOS ACTUALES	MODIFICAR POR
Cuenta de Usuario	: smaldonado	scubas
Descripción	: Sonia Maldonado Cubas	Sonia Cubas Maldonado
Home	: smaldonado	scubas
Grupo Primario	: ejecutivo	-----
Otro Grupo	: marketing	-----

Marque la opción u opciones que permitan crear dicha cuenta:

- a. usermod -c "Sonia Maldonado Cubas" -d /home/scubas -g ejecutivo -G marketing -l scubas smaldonado
- b. usermod -c "Sonia Cubas Maldonado" -d /home/scubas -m -l scubas smaldonado
- c. usermod -c "Sonia Cubas Maldonado" -d /home/scubas -m -g ejecutivo -G marketing smaldonado
- d. usermod -c "Sonia Maldonado Cubas" -d /home/smaldonado -m -g ejecutivo -G marketing -l scubas smaldonado
- e. usermod -c "Sonia Cubas Maldonado" -d /home/scubas -m -G ejecutivo -g marketing -l scubas smaldonado
- f. N.A

5. Cuando se crea una cuenta de usuario. ¿Cuál de las siguientes tareas puede realizar?

- a. Crear otras cuentas de usuario ()
- b. Empaquetar ficheros ()
- c. Cambiar de dueño a un fichero ()
- d. Cambiar la clave del usuario root ()
- e. Eliminar una cuenta de usuario ()
- f. Puede realizar todas las tareas listadas ()

6. La siguiente línea de texto corresponde a:

mysql:x:101:102:MySQL server:/var/lib/mysql:/bin/false

- a. /etc/inittab ()
- b. /etc/group ()
- c. /etc/login.defs ()
- d. /etc/rc.local ()
- e. /etc/shadow ()
- f. /etc/passwd ()

7. Comando que permite obtener la siguiente salida:

Login: s9103846 Name: Gonzales Sanchez Santiago
Directory: /home/sistemas/s9103846 Shell: /bin/false

- a. cat ()
- b. w ()
- c. ls ()
- d. finger ()
- e. passwd ()
- f. useradd ()

Solucionario

1. Del siguiente listado marcar cual permite tener ingreso al sistema

- | | |
|--|--------|
| a. operator:x:11:0:operator:/root:/sbin/nologin | [] |
| b. mysql:x:501:501::/home/mysql:/bin/false | [] |
| c. a0223657:x:1015:10:Melo Perez Brigitte:/home/a0223657:/bin/sh | [x] |
| d. d8552196:x:5543:10:Torres Rios Alinson:/home/d8552196:/bin/bash | [x] |
| e. Irojas:x:7185:10::/home/personal/Irojas:/bin/false | [] |
| f. r_soto:x:7480:10:Raul Soto Garibay:/home/r_soto:/bin/bash | [x] |

2. a. groupadd contadores
groupadd asesores

b. useradd -c "Sonia Vasquez Villar" -g contadores -G asesores svasques
passwd svasques

c. useradd -c "Adrian Gonzales Guisado" -g contadores agonzales
passwd agonzales

d. mkdir /home/asesores
useradd -c "Erika Cárdenas Cuba" -g asesores -s /bin/false ecardenas
passwd ecardenas

e. En el archivo /etc/passwd cada línea representa a una cuenta de usuario, para determinar el total de estos usuarios registrados podemos ejecutar:

wc -l /etc/passwd

Donde el número obtenido representa al total de usuarios.

3. e

4. b

5. b

6. f

7. d

Bibliografía

- [1]. Ball, Hill y Duff, Hoyt (2005) *Red Hat Linux. Fedora 3*. Madrid. Ediciones Anaya Multimedia.
- [2]. Bautts, Tony y Otros (2005) *Linux. Guía para Administradores*. Madrid. Ediciones Anaya Multimedia / O'Reilly.
- [3]. Negus, Christopher (2003) *Red Hat Linux 8*, Madrid. Ediciones Anaya Multimedia.
- [4]. Santos Orcero, David (2008) "Mitos y realidades: Linux y los virus", Revista Todo Linux, Número 90, Pp. 27.

Enlaces

- Baig Viñas, Roger y Aulí Llinás (2003) "Sistema Operativo GNU/Linux Básico" *Formación de Posgrado de la UOC - Máster oficial de Software libre*.
http://www.uoc.edu/masters/oficiales/master_oficial_software_libre/master_oficial_software_libre_materiales.htm
- De Hoyos Marco, Antonio (2005) "GNU/Linux – Administración de Usuarios".
<http://tecnicoslinux.com.ar/web/node/30>
- Kirch, Olaf y Dawson, Ferry (2002) "Guía de Administración de Redes con Linux". *O'Reilly (printed version) (c) 2000 O'Reilly & Associates. Proyecto LuCAS por la traducción al español*.
<http://es.tldp.org/Manuales-LuCAS/GARL2/garl2/>
- Red Hat, Inc (2005) "Red Hat Enterprise Linux 4 - Introducción a la administración de sistemas"
<http://www.redhat.com/docs/manuals/enterprise/RHEL-4-Manual/es/admin-guide/>
- Red Hat, Inc (2005) "Red Hat Enterprise Linux 4 - Manual de Referencia"
<http://www.redhat.com/docs/manuals/enterprise/RHEL-4-Manual/es/ref-guide/>

UNIDAD IV

PERMISOS DE FICHEROS

La unidad tiene como propósito gestionar los permisos de los ficheros en los sistemas GNU/Linux; así mismo, en los usuarios y los grupos que tienen control sobre los ficheros valorando la importancia de los conocimientos para su desarrollo académico.

Comprende:

- Conceptos básicos
- Tipos de permisos
- Visualizar permisos
- Comandos
- Cambiando propietarios y grupos

Lección 9

Permisos

9.1. Conceptos básicos

Normalmente cuando deseamos ingresar a un directorio o editar un archivo nos muestra el siguiente mensaje:

```
[root@fisct ~]$ cd /root  
bash: /root: Permission denied
```

El ejemplo anterior nos muestra una de las características de GNU/Linux que es la seguridad. GNU/Linux, como UNIX, es un sistema multiusuario y los permisos para tener acceso a los ficheros presentan una solución para proteger la integridad del sistema ante cualquier daño [3].

Para tener acceso deberá ejecutar la siguiente orden:

```
[root@fisct ~]# su -  
Contraseña:  
[root@fisct ~]# cd /root
```

Para ello deberá conocer la contraseña de root para tener acceso completo al sistema.

El sistema de permisos en GNU/Linux se basa en un esquema de usuarios/grupos que lo convierte en la base principal de la seguridad en GNU/Linux, a estos usuarios y grupos se les asignan distintos derechos sobre los ficheros (archivos y directorios) [1] [4].

9.2. Tipos de permisos

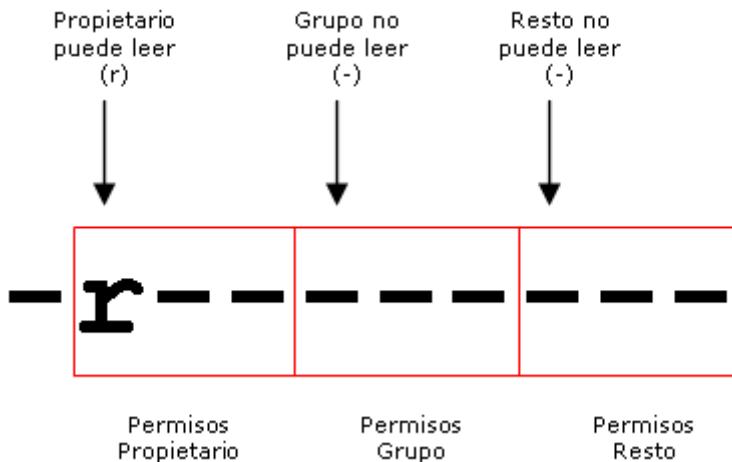
La gestión de los permisos en los sistemas GNU/Linux, los usuarios y los grupos tienen el control sobre los archivos y los directorios. Esto se realiza mediante un esquema de tres tipos de permisos que son:

9.2.1. Permiso de lectura

Cuando un usuario tiene permiso de lectura sobre un archivo significa que puede leerlo o visualizarlo, mediante una aplicación o comandos. Por ejemplo, si tenemos permiso de lectura sobre el archivo examen.txt, significa que podemos ver el contenido del archivo. Si el usuario no tiene permiso de lectura, no podrá ver el contenido del archivo.

Cuando un usuario tiene permiso de lectura sobre un directorio, significa que puede visualizar el contenido de la carpeta, es decir, puede ver los archivos y directorios que contiene, utilizando el comando 'ls' o con un explorador de archivos como Konqueror. Si el usuario no tiene permiso de lectura sobre el directorio, no podrá ver su contenido.

El permiso de lectura se simboliza con la letra 'r' del inglés 'read'.

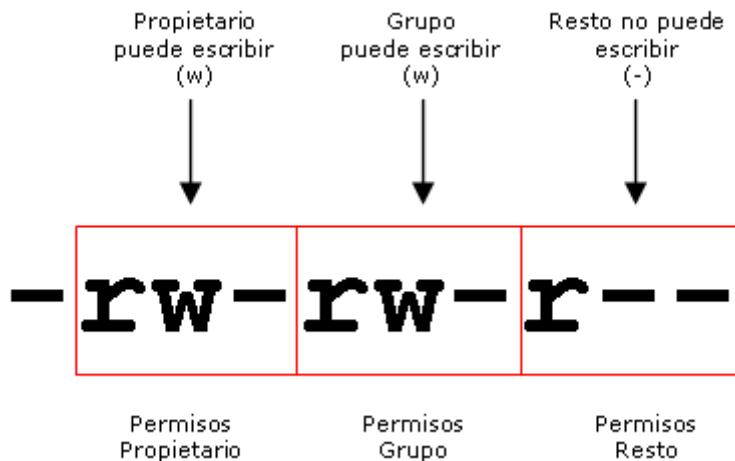


9.2.2. Permiso de escritura

Cuando un usuario tiene permiso de escritura sobre un archivo significa que puede modificar su contenido, e incluso borrarlo. También le da derecho a cambiar los permisos del archivo mediante el comando **chmod** así como cambiar su propietario y el grupo propietario mediante el comando **chown**. Si el usuario no tiene permiso de escritura, no podrá modificar el contenido del archivo.

Cuando un usuario tiene permiso de escritura sobre un directorio, significa que puede modificar el contenido del directorio, es decir, puede crear y eliminar archivos y otros directorios dentro de ella. Si el usuario no tiene permiso de escritura sobre el directorio, no podrá crear ni eliminar archivos ni directorios dentro de ella.

El permiso de escritura se simboliza con la letra 'w' del inglés 'write'.



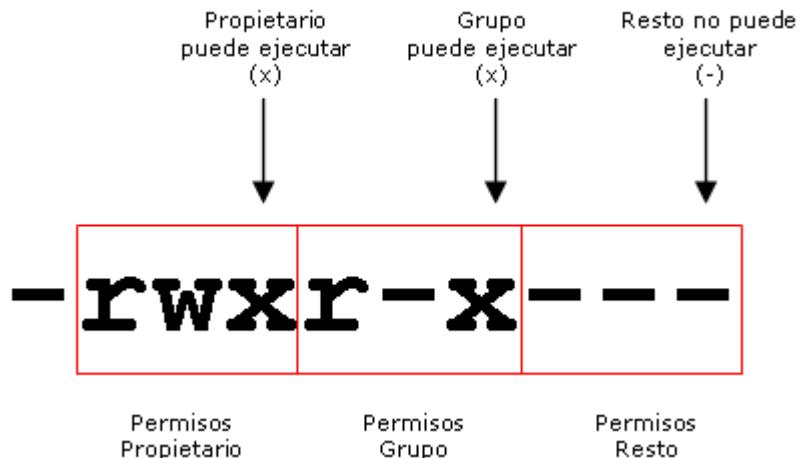
9.2.3. Permiso de ejecución

Cuando un usuario tiene permiso de ejecución sobre un archivo significa que puede ejecutarlo. Si el usuario no dispone de permiso de ejecución, no podrá ejecutarlo aunque sea una aplicación.

Los únicos archivos ejecutables son las aplicaciones y los archivos de comandos (scripts). Si tratamos de ejecutar un archivo no ejecutable, mostrará errores.

Cuando un usuario tiene permiso de ejecución sobre un directorio, significa que puede entrar en ella, usando el comando '**cd**'. Si no dispone del permiso de ejecución significa que no puede ingresar al directorio.

El permiso de ejecución se simboliza con la letra 'x' del inglés 'eXecute'.



9.3. Visualizar los permisos

Para ver los permisos de los archivos y directorios es necesario ejecutar el siguiente comando:

```
[root@fisct ~]# ls -l /boot/grub
```

Este comando nos dará una salida similar a la siguiente:

```
-rw-r--r-- 1 root root    63 dic 30 05:04 device.map
-rw-r--r-- 1 root root  7584 dic 30 05:04 e2fs_stage1_5
-rw-r--r-- 1 root root  7456 dic 30 05:04 fat_stage1_5
-rw-r--r-- 1 root root  6720 dic 30 05:04 ffs_stage1_5
-rw----- 1 root root   923 ene 12 11:46 grub.conf
-rw-r--r-- 1 root root  6720 dic 30 05:04 iso9660_stage1_5
-rw-r--r-- 1 root root  8192 dic 30 05:04 jfs_stage1_5
lrwxrwxrwx 1 root root   11 dic 30 05:04 menu.lst -> ./grub.conf
-rw-r--r-- 1 root root  6880 dic 30 05:04 minix_stage1_5
-rw-r--r-- 1 root root  9248 dic 30 05:04 reiserfs_stage1_5
-rw-r--r-- 1 root root 55808 mar 12 2009 splash.xpm.gz
-rw-r--r-- 1 root root    512 dic 30 05:04 stage1
-rw-r--r-- 1 root root 104956 dic 30 05:04 stage2
-rw-r--r-- 1 root root  7072 dic 30 05:04 ufs2_stage1_5
-rw-r--r-- 1 root root  6272 dic 30 05:04 vstafs_stage1_5
-rw-r--r-- 1 root root  8864 dic 30 05:04 xfs_stage1_5
```

La descripción de la salida es la siguiente:

Con la siguiente línea interpretamos la información así:

```
- rw- r-- r-- 1 root root 7584 dic 30 05:04 e2fs_stage1_5
↑ ↑   ↑   ↑   ↑   ↑   ↑   ↑   ↑   ↑   ↑
1 2   3   4   5 6   7   8   9   10  11
```

1 : Tipo de archivo	= es un archivo regular
2 : Permisos	= los permisos para el propietario son de lectura y escritura
3 : Permisos	= el grupo tiene permiso de sólo lectura
4 : Permisos	= los otros usuarios tienen el permiso de sólo lectura
5 : Enlace Físico	= tiene un enlace físico
6 : Propietario	= el usuario raul es el propietario o dueño de este archivo
7 : Grupo	= este archivo pertenece al grupo raul
8 : Tamaño	= su tamaño es de 246417 bytes
9 : Fecha	= fue creado o modificado el 03 de marzo de 2005
10 : Hora	= a 13:13 horas
11 : Nombre	= el archivo se llama agenda

Los permisos están asignados en grupos de 3 (rwx) y corresponde al: propietario (**owner**: dueño del archivo o directorio), grupo (**group**: grupo del archivo o directorio) y otros (**others**: otro usuario diferente del propietario).



9.4. Comandos

GNU/Linux dispone de 3 comandos que permite cambiar los permisos, el propietario y el grupo de un archivo y/o directorio respectivamente:

- **Comando chmod** : se utiliza para cambiar los permisos del fichero
Sintaxis: `chmod [opciones] [permisos] [fichero]`
- **Comando chown** : se utiliza para cambiar el propietario del fichero
Sintaxis: `chown [opciones] [nuevo propietario] [fichero]`
- **Comando chgrp** : utilizado para cambiar el grupo del fichero
Sintaxis: `chgrp [opciones] [nuevo grupo] [fichero]`

Opciones:

- R** : Indica recursividad, aplicará los permisos a todos los ficheros contenidos en el directorio.
- f** : No muestra mensajes de error sobre ficheros cuyos permisos no se pueden cambiar.

Para cambiar los permisos se puede hacer de 2 maneras:

9.4.1. Cambio de permisos utilizando caracteres

Para poder utilizar cambiar permisos basado en caracteres tomemos en cuenta la siguiente lista con su respectiva correspondencia:

Descripción	Símbolo	Descripción
Identidades	u	Es el usuario propietario del archivo o directorio
	g	Es el grupo al que pertenece el archivo o directorio
	o	Otros usuarios, ni el propietario ni su grupo
	a	Todo el mundo: propietario, grupo y otros
Permisos	r	Acceso de lectura
	w	Acceso de escritura
	x	Acceso de ejecución
Acciones	+	Añade los permisos
	-	Elimina los permisos
	=	el único permiso

Sintaxis: chmod {a,u,g,o} {+,-} {r,w,x} <fichero>

Para los siguientes ejemplos deberá crear el archivo agenda:

Ejemplo	Descripción	Resultado
# touch linux	creamos el archivo linux	agenda
# chmod a-rwx linux	quitamos todos los permisos al archivo linux	-----
# chmod u+rwx linux	añadimos todos los permisos para el propietario	rwx-----
# chmod g+x linux	añadimos el permiso de ejecución para el grupo	rwx--x---
# chmod o+r linux	añadimos el permiso de lectura para los otros usuarios	rwx--xr--
# chmod u-rw linux	eliminamos los permisos de lectura y escritura para el propietario	--x--xr--
# chmod a=r linux	establecemos como único permiso de lectura para los 3 grupos	r--r--r--
# chmod a=rx linux	establecemos los permisos de lectura y ejecución para los 3 grupos	r-xr-xr-x
# chmod a=- linux	quitamos todos los permisos	-----
# chmod u+rx,o+x linux	añadimos los permisos de lectura y ejecución al propietario y ejecución a otros	r-x-----x
# chmod g+rx,o-x linux	añadimos permiso de lectura y ejecución al grupo y eliminamos permiso de ejecución a otros	r-xr-x---
# chmod ug+wx,o-x linux	añadimos permiso de escritura y ejecución al propietario y grupo, y eliminamos permiso de ejecución a otros	rwxrwx---
# chmod a=rw linux	permite a cualquiera modificar el contenido e incluso eliminar el archivo	rw-rw-rw-

Si cambiamos los permisos a un directorio y deseamos que estos permisos tengan efecto sobre todos sus subdirectorios y archivos sólo deberemos añadir la opción -R.

Ejemplo:

```
[root@fisct ~]# chmod -R a=rw DIRECTORIO
```

9.4.2. Cambio de permisos utilizando números o Modo Octal

Cada permiso tiene asignado un valor numérico, incluso cuando el permiso no está activo. Para poder utilizar los números tendremos que tener en cuenta la siguiente tabla con sus respectivos valores:

r	=	4 (lectura)
w	=	2 (escritura)
x	=	1 (ejecución)
-	=	0 (sin permisos)

Cuando asignamos los permisos utilizando números debemos tener en cuenta que primero se sumarán los valores y dicho resultado será el que se coloque. Se muestra la tabla con los siguientes valores:

Valor	Permisos	Descripción
0	---	El valor cero significa que no se han asignado permisos
1	--x	sólo se ha asignado el de ejecución
2	-w-	sólo permiso de escritura
3	-wx	permisos de escritura y ejecución
4	r--	sólo permiso de lectura
5	r-x	permisos de lectura y ejecución
6	rw-	permisos de lectura y escritura
7	rwx	permisos: lectura, escritura y ejecución

Los permisos por números se asignan en grupos de 3, es decir, para el propietario-grupo-otros, no es factible asignar solo para uno o dos de ellos.

Ejemplos:

- **rw----- (600)** Sólo el propietario tiene el derecho de leer y escribir.
- **rw-r--r-- (644)** Sólo el propietario tiene los permisos de leer y escribir; el grupo y los demás sólo pueden leer.
- **rwx----- (700)** Sólo el propietario tiene los derechos de leer, escribir y ejecutar el archivo e ingresar al directorio.
- **rwxr-xr-x (755)** El propietario tiene los derechos de leer, escribir y ejecutar; el grupo y los demás sólo pueden leer y ejecutar.
- **rwx--x--x (711)** El propietario tiene los derechos de lectura, escritura y ejecución; el grupo y los demás sólo pueden ejecutar.
- **rw-rw-rw- (666)** Todo el mundo puede leer y escribir en el archivo.
- **rwxrwxrwx (777)** Todo el mundo puede leer, escribir y ejecutar.

En binario, las combinaciones representan el tipo de permisos. El bit más a la derecha (menos significativo) se refiere al permiso de ejecución (1=activar y 0=desactivar). El

bit central se refiere al permiso de escritura y el bit más a la izquierda se refiere al permiso de lectura. La siguiente tabla muestra las 8 combinaciones posibles:

Código Binario Permisos efectivos

0	0 0 0	- - -
1	0 0 1	- - x
2	0 1 0	- w -
3	0 1 1	- w x
4	1 0 0	r - -
5	1 0 1	r - x
6	1 1 0	r w -
7	1 1 1	r w x

Crear el archivo **foto.png** para realizar los siguientes ejercicios:

```
[root@fisct ~]# touch foto.png
```

Ejemplo	Descripción	Resultado
# touch foto.png	creamos el archivo foto.png	foto.png
# chmod 000 foto.png	quitamos todos los permisos al archivo foto.png	-----
# chmod 700 foto.png	añadimos todos los permisos para el propietario	-rwx----
# chmod 710 foto.png	añadimos el permiso de ejecución para el grupo	-rwx-x--
# chmod 714 foto.png	añadimos el permiso de lectura para los otros usuarios	-rwx-xr--
# chmod 114 foto.png	eliminamos los permisos de lectura y escritura para el propietario	---x-xr--
# chmod 444 foto.png	establecemos como único permiso de lectura para el dueño, grupo y demás usuarios	-r--r--r--
# chmod 555 foto.png	establecemos los permisos de lectura y ejecución para el dueño, grupo y demás usuarios	-r-xr-xr-x
# chmod 000 foto.png	quitamos todos los permisos	-----
# chmod 501 foto.png	añadimos los permisos de lectura y ejecución al propietario y ejecución a otros	-r-x-----x
# chmod 550 foto.png	añadimos permiso de lectura y ejecución al grupo y eliminamos permiso de ejecución a otros	-r-xr-x---
# chmod 770 foto.png	añadimos permiso de escritura y ejecución al propietario y grupo, y eliminamos permiso de ejecución a otros	-rwxrwx---
# chmod 666 foto.png	permite a cualquiera modificar el contenido e incluso eliminar el archivo	-rw-rw-rw-

9.5. Cambiando Propietarios y Grupos

Otra de los puntos a la hora de establecer permisos es la necesidad de poder cambiar el propietario y grupo del archivo o directorio. Para hacer esta operación debe estar

como usuario **root**, los usuarios y grupos que utilizará deben haber sido creados previamente.

9.5.1. Cambiando el propietario

Utilizamos el comando **chown** para cambiar el propietario:

```
# chown sonia agenda          # estamos cambiando el propietario del  
archivo, ahora el usuario sonia será el propietario del archivo agenda
```

```
# chown jlopez config.php      # el usuario jlopez será el propietario del  
archivo config.php
```

Si vamos a cambiar el propietario de un directorio y con todos sus subdirectorios y archivos en forma recursiva utilizaremos la opción **-R**

```
# chown -R webmaster documentos  # el usuario webmaster será el nuevo  
propietario de todos los archivos y subdirectorios que estén dentro del directorio  
documentos
```

9.5.2. Cambiando el grupo

Utilizamos el comando **chgrp** para el cambiar el grupo:

```
# chgrp users agenda          # estamos cambiando el propietario del archivo,  
ahora el archivo agenda será del grupo users
```

```
# chgrp srojas config.php     # el archivo config.php será del grupo srojas
```

Si vamos a cambiar el grupo de un directorio y con todos sus subdirectorios y archivos en forma recursiva utilizaremos la opción **-R**

```
# chgrp -R clases documentos   # todos los archivos y sub directorios del directorio  
documento serán del grupo clases
```

9.5.3. Cambiar usuario propietario y grupo propietario

Para poder cambiar el usuario propietario y el grupo propietario de un archivo o directorio se utiliza el comando **chown**. Para ello hay que **disponer de permisos de escritura** sobre el archivo o directorio.

La sintaxis del comando es:

```
# chown nuevo_usuario[.nuevo_grupo] nombre_archivo
```

Resumen

En esta unidad, se abordó el tema de permisos en los ficheros en el sistema GNU/Linux el cual se basa en un esquema de usuarios/grupos que lo convierte en la base principal de la seguridad en GNU/Linux. A estos usuarios y grupos se les asignan distintos derechos sobre los archivos y directorios.

Lectura

Hackers, crackers, seguridad y libertad

Manuel Castells

Profesor senior del Internet Interdisciplinary Institute (IN3) de la UOC

Los hackers y su cultura son una de las fuentes esenciales de la invención y continuo desarrollo de Internet. Los hackers no son lo que los medios de comunicación o los gobiernos dicen que son. Son, simplemente, personas con conocimientos técnicos informáticos cuya pasión es inventar programas y desarrollar formas nuevas de procesamiento de información y comunicación electrónica (Levy, 1984; Raymond, 1999). Para ellos, el valor supremo es la innovación tecnológica informática. Y, por tanto, necesitan también libertad. Libertad de acceso a los códigos fuente, libertad de acceso a la red, libertad de comunicación con otros hackers, espíritu de colaboración y de generosidad (poner a disposición de la comunidad de hackers todo lo que se sabe, y, en reciprocidad, recibir el mismo tratamiento de cualquier colega). Algunos hackers son políticos y luchan contra el control de los gobiernos y de las corporaciones sobre la red, pero la mayoría no lo son, lo importante para ellos es la creación tecnológica. Se movilizan, fundamentalmente, para que no haya cortapisas a dicha creación. Los hackers no son comerciales, pero no tienen nada contra la comercialización de sus conocimientos, con tal de que las redes de colaboración de la creación tecnológica sigan siendo abiertas, cooperativas y basadas en la reciprocidad.

La cultura hacker se organiza en redes de colaboración en Internet, aunque de vez en cuando hay algunos encuentros presenciales. Distintas líneas tecnológicas se agrupan en torno a grupos cooperativos, en los cuales se establece una jerarquía tecnológica según quiénes son los creadores de cada programa original, sus mantenedores y sus contribuidores. La comunidad suele reconocer la autoridad de los primeros innovadores, como es el caso de Linus Torvalds en la comunidad Linux. Pero sólo se reconoce la autoridad de quien la ejerce con prudencia y no la utiliza para su beneficio personal.

El movimiento hacker más político (en términos de política de libertad tecnológica) es el creado por Richard Stallman, un programador de MIT, que constituyó en los años ochenta la Free Software Foundation para defender la libertad de acceso a los códigos de UNIX cuando ATT trató de imponer sus derechos de propiedad sobre UNIX, el sistema operativo más avanzado y más compatible de su tiempo, y sobre el que se ha fundado en buena parte la comunicación de los ordenadores en la red. Stallman, que aprendió el valor de la libertad en el movimiento de libre expresión en sus tiempos de estudiante en Berkeley, sustituyó el copy right por el copy left. Es decir, que cualquier programa publicado en la red por su Fundación podía ser utilizado y modificado bajo licencia de la Fundación bajo una condición: difundir en código abierto las modificaciones que se fueran efectuando. Sobre esa base, desarrolló un nuevo sistema operativo, GNU, que sin ser Unix, podía utilizarse como UNIX. En 1991, un estudiante de 21 años de la Universidad de Helsinki, Linus Torvalds, diseñó su propio UNIX kernel para su PC 386 sobre la base de Fundación. Y, siguiendo las reglas del juego, publicó la fuente de su código en la red, solicitando ayuda para perfeccionarlo. Cientos de programadores espontáneos se pusieron a la tarea, desarrollando así el sistema operativo Linux (que recibió ese nombre del administrador del sistema en la Universidad de Helsinki, puesto que el nombre que Torvalds le había dado era el de Freix), considerado hoy en día el más avanzado del mundo, sobre todo para ordenadores en Internet, y la única alternativa actual a los programas de Microsoft. Linux cuenta en la actualidad con más de 30 millones de usuarios y está siendo promocionado por los gobiernos de Francia, de Brasil, de la India, de Chile, de China, entre otros, así como por grandes empresas como IBM. Siempre en código abierto y sin derechos de propiedad sobre él.

El filósofo finlandés Pekka Himanen (www.hackerethic.org) argumenta convincentemente que la cultura hacker es la matriz cultural de la era de la información, tal y como la ética protestante fue el sistema de valores que coadyuvó decisivamente al desarrollo del capitalismo, según el análisis clásico de Max Weber. Naturalmente, la mayoría de los capitalistas no era protestante ni la mayoría de los actores de la sociedad de la información es hacker. Pero lo que esto significa es lo siguiente: una gran transformación tecnoeconómica necesita un caldo de cultivo en un sistema de valores nuevo que motive a la gente para hacer lo que hace. En el caso del capitalismo, fue la ética del trabajo y de la acumulación de capital en la empresa como forma de salvación personal (lo cual, desde luego, no impidió, sino que justificó, la explotación de los trabajadores).

En la era de la información, la matriz de todo desarrollo (tecnológico, económico, social) está en la innovación, en el valor supremo de la innovación que, potenciada por la revolución tecnológica informacional, incrementa exponencialmente la capacidad de generación de riqueza y de acumulación de poder. Pero innovar no es un valor obvio. Debe estar asociado a una satisfacción personal, del tipo que sea, ligado al acto de la innovación. Eso es la cultura hacker, según Himanen. El placer de crear por crear. Y eso mueve el mundo, sobre todo el mundo en que la creación cultural, tecnológica, científica y también empresarial, en su aspecto no crematístico, se convierte en fuerza productiva directa por la nueva relación tecnológica entre conocimiento y producción de bienes y servicios. Se podría argumentar que, así definido, hay hackers en todas partes y no sólo en la informática. Y ése es, en realidad, el argumento de Himanen: que todo el mundo puede ser hacker en lo que hace y que cualquiera que esté movido por la pasión de crear en su actividad propia está motivado por una fuerza superior a la de la ganancia económica o la satisfacción de sus instintos. Lo que ocurre es que la innovación tecnológica informática tiene el piñón directo sobre la rueda del cambio en la era de la información, de ahí que la cultura hacker se manifieste de forma particularmente espectacular en las tecnologías de información y en Internet.

En realidad, los hackers han sido fundamentales en el desarrollo de Internet. Fueron hackers académicos quienes diseñaron los protocolos de Internet. Un hacker, Ralph Tomlinson, trabajador de la empresa BBN, inventó el correo electrónico en 1970, para uso de los primeros internautas, sin comercialización alguna. Hackers de los Bell Laboratories y de la Universidad de Berkeley desarrollaron UNIX. Hackers estudiantes inventaron el módem. Las redes de comunicación electrónica inventaron los tablones de anuncio, los chats, las listas electrónicas y todas las aplicaciones que hoy estructuran Internet. Y Tim Berners-Lee y Roger Cailliau diseñaron el browser/editor World Wide Web, por la pasión de programar, a escondidas de sus jefes en el CERN de Ginebra, en 1990, y lo difundieron en la red sin derechos de propiedad a partir de 1991. También el browser que popularizó el uso del World Wide Web, el Mosaic, fue diseñado en la Universidad de Illinois por otros dos hackers (Marc Andreessen y Eric Bina) en 1992. Y la tradición continúa: en estos momentos, dos tercios de los servidores de web utilizan Apache, un programa servidor diseñado y mantenido en software abierto y sin derechos de propiedad por una red cooperativa.

En una palabra, los hackers informáticos han creado la base tecnológica de Internet, el medio de comunicación que constituye la infraestructura de la sociedad de la información. Y lo han hecho para su propio placer, o, si se quiere, por el puro goce de crear y compartir la creación y la competición de la creación. Ciertamente, unos pocos de entre ellos también se hicieron ricos como empresarios, pero mediante aplicaciones de sus innovaciones, no mediante la apropiación de la innovación cooperativa en su propio beneficio (aunque el caso de Andreessen es menos claro, en este sentido). Otros obtuvieron buenos puestos de trabajo, pero sin ceder en sus principios como hackers. También hubo quien se hizo famoso, como Linus Torvalds, pero su fama vino de su reconocimiento de la comunidad de hackers, que implica el respeto a sus reglas de libertad y cooperación. Los más permanecieron anónimos para el mundo y llevan y

I llevaron una vida modesta. Pero obtuvieron, mediante su práctica de innovación cooperativa, la más alta recompensa a la que aspira un hacker, el reconocimiento como tal por parte de la única autoridad que puede otorgar dicha distinción: la comunidad global de hackers, fuente esencial de innovación en la era de la información.

En los márgenes de la comunidad hacker se sitúan los crackers. Los crackers, temidos y criticados por la mayoría de hackers, por el desprestigio que les supone ante la opinión pública y las empresas, son aquellos que utilizan sus conocimientos técnicos para perturbar procesos informáticos.

Hay muy distintos tipos de crackers, pero no considero entre ellos a aquellos que penetran en ordenadores o redes de forma ilegal para robar: éstos son ladrones de guante blanco, una vieja tradición criminal. Muchos crackers pertenecen a la categoría de script kiddies, es decir, bromistas de mal gusto, muchos de ellos adolescentes, que penetran sin autorización en sistemas o crean y difunden virus informáticos para sentir su poder, para medirse con los otros, para desafiar al mundo de los adultos y para chulear con sus amigos o con sus referentes en la red. La mayoría de ellos tiene conocimientos técnicos limitados y no crea ninguna innovación, por lo que son, en realidad, marginales al mundo hacker. Otros crackers, más sofisticados, penetran en sistemas informáticos para desafiar personalmente a los poderes establecidos, por ejemplo, a Microsoft o las grandes empresas. Y algunos utilizan su capacidad tecnológica como forma de protesta social o política, como expresión de su crítica al orden establecido. Ellos son quienes se introducen en sistemas militares, administraciones públicas, bancos o empresas para reprocharles alguna fechoría. Entre los ataques de crackers con motivación política hay que situar los practicados por movimientos políticos o por servicios de inteligencia de los gobiernos, como la guerra informática desarrollada entre los crackers islámicos e israelíes o entre los prochechenos y los servicios rusos.

En suma, en la medida en que los sistemas informáticos y las comunicaciones por Internet se han convertido en el sistema nervioso de nuestras sociedades, la interferencia con su operación a partir de una capacidad técnica de actuación en la red es un arma cada vez más poderosa, que puede ser utilizada por distintos actores y con distintos fines. Éstas son las acciones de los crackers, que deben ser absolutamente deslindados de los hackers, a cuya constelación pertenecen, pero con quienes no se confunden.

La vulnerabilidad de los sistemas informáticos plantea una contradicción creciente entre seguridad y libertad en la red. Por un lado, es obvio que el funcionamiento de la sociedad y sus instituciones y la privacidad de las personas no puede dejarse al albur de cualquier acción individual o de la intromisión de quienes tienen el poder burocrático o económico de llevarla a cabo. Por otro lado, como ocurre en la sociedad en general, con el pretexto de proteger la información en la red se renueva el viejo reflejo de control sobre la libre comunicación.

El debate sobre seguridad y libertad se estructura en torno a dos polos: por un lado, la regulación político-jurídica de la red; por otro, la autoprotección tecnológica de los sistemas individuales. Naturalmente, hay fórmulas intermedias, pero, en general, dichas fórmulas mixtas tienden a gravitar hacia la regulación institucional de la comunicación electrónica. Quienes defienden la capacidad de autorregulación de la red argumentan que existen tecnologías de protección que son poco vulnerables, sobre todo cuando se combinan los fire walls (o filtros de acceso) de los sistemas informáticos con las tecnologías de encriptación, que hacen muy difíciles de interceptar los códigos de acceso y el contenido de la comunicación. Es así como están protegidos los ordenadores del Pentágono, de los bancos suizos o de Scotland Yard. La mayor parte de las instituciones de poder y de las grandes empresas tiene sistemas

de seguridad a prueba de cualquier intento de penetración que no cuente con capacidad tecnológica e informática similar. Ciertamente hay una carrera incesante entre sistemas de ataque informático y de protección de éstos, pero por esto mismo, el corazón de dichos sistemas es poco vulnerable para el comunitario de los hackers.

Ahora bien, al estar los sistemas informáticos conectados en red, la seguridad de una red depende en último término de la seguridad de su eslabón más débil, de forma que la capacidad de penetración por un nodo secundario puede permitir un ataque a sus centros más protegidos. Esto fue lo que ocurrió en el año 2000 cuando los crackers se introdujeron en el sistema de Microsoft y obtuvieron códigos confidenciales, a partir de la penetración en el sistema personal de un colaborador de Microsoft que tenía acceso a la red central de la empresa. Es manifiestamente imposible proteger el conjunto de la red con sistemas de fire walls y encriptación automática. Por ello, sólo la difusión de la capacidad de encriptación y de autoprotección en los sistemas individuales podría aumentar la seguridad del sistema en su conjunto. En otras palabras, un sistema informático con capacidad de computación distribuida en toda la red necesita una protección igualmente distribuida y adaptada por cada usuario a su propio sistema. Pero eso equivale a poner en manos de los usuarios el poder de encriptación y autoprotección informática. Algo que rechazan los poderes políticos con el pretexto de la posible utilización de esta capacidad por los criminales (en realidad, las grandes organizaciones criminales tienen la misma capacidad tecnológica y de encriptación que los grandes bancos). En último término, la negativa de las administraciones a permitir la capacidad de encriptación y de difusión de tecnología de seguridad entre los ciudadanos conlleva la creciente vulnerabilidad de la red en su conjunto, salvo algunos sistemas absolutamente aislados y, en última instancia, desconectados de la red.

De ahí que gobiernos y empresas busquen la seguridad mediante la regulación y la capacidad represiva de las instituciones más que a través de la autoprotección tecnológica de los ciudadanos. Es así como se reproduce en el mundo de Internet la vieja tensión entre seguridad y libertad.

Fuente [2]

Autoevaluación

1. Crear una cuenta de usuario para "José Carrillo Chávez". Abrir una sesión con la cuenta creada.
2. Crear un directorio llamado nuevo, ingrese y crear 9 archivos (archiv1, archiv2, etc.) utilizando el comando touch.
Quitarle todos los permisos con el comando "chmod a-rwx archiv*"
3. Modificar los permisos usando el operador '=' del 'chmod', para que queden de la siguiente manera:

archiv1 -rwx-----	\$ chmod u=rwx,go= archiv1
archiv2 -rw-----	\$
archiv3 -rwxrwxrwx	\$ chmod a=rwx archiv3
archiv4 -rwxrw-r--	\$
archiv5 -rwxr-----	\$
archiv6 -r-xrw-r--	\$
archiv7 -r-----x	\$
archiv8 -rw-r--r--	\$
archiv9 -rw-rw-r--	\$

4. Modificar los permisos de los archivos anteriores utilizando los operadores + y - del 'chmod' para que queden de la siguiente manera:

archiv1 -rwx---r--	\$ chmod o+r archiv1
archiv2 -r-----	\$
archiv3 -rw-rw-rw-	\$ chmod a-x archiv3
archiv4 -rwx-w----	\$
archiv5 -rwx----wx	\$
archiv6 -rwxrw----	\$
archiv7 -rw---x-w-	\$
archiv8 -----r--	\$
archiv9 -rwx-----	\$

5. Crear 9 archivos (num1, num2, etc.) utilizando el comando touch.

6. Sobreescribir los permisos utilizando el comando chmod con argumento numérico (octal) para que queden de la siguiente manera:

num1 -r---w---x	\$ chmod 421 num1
num2 -----	
num3 -rwxrwxrwx	\$ chmod 777 num3
num4 -r-xrw-r--	
num5 -rwxr-----	
num6 -rw-r--r--	
num7 -rw-r--r-x	
num8 -rwxrw-r--	
num9 -rwx-----	

7. Con una sola instrucción, quitar permisos de lectura, escritura y ejecución para "otros" a todos los archivos (num) utilizados en el ejercicio anterior (6).
8. Crear una cuenta de usuario para "Rosa Pérez Sánchez", luego abrir una sesión con esta cuenta. Crear el directorio documentos y quitarle todos los permisos de ejecución.

Explicar qué pasa al intentar entrar al directorio con el comando cd.

Explicar el significado de los permisos r, w y x para directorios.

Explicar el significado de los permisos r, w y x para archivos.

9. Utilizando los comandos chown y chgrp, intentar cambiar de propietario a edominguez y el grupo a soporte en el archivo "num3". ¿Cuál es el problema?

Solucionario

1. useradd -c "José Carrillo Chávez" jcarrilloc
passwd jcarrilloc
2. # mkdir nuevo
cd nuevo
touch archiv1 archiv2 archiv3 archiv4 archiv5 archiv6 archiv7 archiv8 archiv9
3. Modificar los permisos usando el operador '=' del 'chmod', para que queden de la siguiente manera:

archiv1 -rwx-----	\$ chmod u=rwx,go= archiv1
archiv2 -rw-----	\$ chmod u=rw archiv2
archiv3 -rwxrwxrwx	\$ chmod a=rwx archiv3
archiv4 -rwxrw-r--	\$ chmod u=rwx,g=rw,o=r archiv4
archiv5 -rwxr-----	\$ chmod u=rwx,g=r archiv5
archiv6 -r-xrw-r--	\$ chmod u=rx,g=rw,o=r archiv6
archiv7 -r-----x	\$ chmod u=r,o=x archiv7
archiv8 -rw-r--r--	\$ chmod u=rw,g=r,o=r archiv8
archiv9 -rw-rw-r--	\$ chmod u=rw,g=rw,o=r archiv9

4. Modificar los permisos de los archivos anteriores utilizando los operadores + y - del 'chmod' para que queden de la siguiente manera:

archiv1 -rwx---r--	\$ chmod o+r archiv1
archiv2 -r-----	\$ chmod u+r
archiv3 -rw-rw-rw-	\$ chmod a-x archiv3
archiv4 -rwx-w----	\$ chmod g-w,g+r
archiv5 -rwx----wx	\$ chmod gu-r
archiv6 -rwxrw----	\$ chmod u+w,o-r
archiv7 -rw---x-w-	\$ chmod ug+w,g+x,o-x
archiv8 -----r--	\$ chmod u-rw,g-r
archiv9 -rwx-----	\$ chmod u+x,g-rw,o-r

5. # touch num1 num2 num3 num4 num5 num6 num7 num8 num9

6. Sobreescribir los permisos utilizando el comando chmod con argumento numérico (octal) para que queden de la siguiente manera:

num1 -r---w---x	\$ chmod 421 num1
num2 -----	\$ chmod 000 num2
num3 -rwxrwxrwx	\$ chmod 777 num3
num4 -r-xrw-r--	\$ chmod 564 num4
num5 -rwxr----	\$ chmod 740 num5
num6 -rw-r--r--	\$ chmod 644 num6
num7 -rw-r--r-x	\$ chmod 645 num7
num8 -rwxrw-r--	\$ chmod 764 num8
num9 -rwx-----	\$ chmod 700 num9

7. chmod 000 num*
8. userdad -c "Rosa Pérez Sánchez" rperez
passwd rperez

Para apertura una sesión presione la combinación de teclas CTRL+ALT+F3, luego ingrese con el usuario rperez y la clave asignada. Una vez que ingrese crear el directorio documentos.

mkdir documentos

Para quitar todos los permisos al directorio documentos utilizar:

chmod 000 documentos

a. El usuario no podrá ingresar al directorio documentos porque no cuenta con los permisos suficientes.

b. Para los directorios:

r = permite leer el contenido

w = permite crear ficheros dentro del directorio

x = permite ingresar al directorio

c. Para los archivos:

r = permite leer el contenido de los archivos

w = permite modificar el contenido de los archivos

x = En caso de ser un archivo script este podrá ser ejecutado.

9. El usuario no dispone de los permisos para cambiar de propietario ni grupo a los ficheros.

Bibliografía

- [1]. Bautts, Tony y Otros (2005) *Linux. Guía para Administradores*. Madrid. Ediciones Anaya Multimedia / O'Reilly.
- [2]. Castells, Manuel. (2001) "Hackers, crackers, seguridad y libertad". Publicado por la Universidad Oberta de Catalunya en <http://www.uoc.edu/inaugural01/esp/hackers.html>. Consultado el 22 de enero de 2009.
- [3]. Kalle, Mathias y Welsh, Matt (2006) *Guía de Referencia y Aprendizaje LINUX*. 2º. Ed. Madrid, Ediciones Anaya Multimedia / O'Reilly.
- [4]. Negus, Christopher (2003) *Red Hat Linux 8*, Madrid. Ediciones Anaya Multimedia.

Enlaces

- Baig Viñas, Roger y Aulí Llinás (2003) "Sistema Operativo GNU/Linux Básico" *Formación de Posgrado de la UOC - Máster oficial de Software libre*. http://www.uoc.edu/masters/oficiales/master_oficial_software_libre/master_oficial_software_libre_materiales.htm
- De Hoyos Marco, Antonio (2005) "GNU/Linux – Administración de Usuarios". <http://tecnicoslinux.com.ar/web/node/30>
- Kirch, Olaf y Dawson, Ferry (2002) "Guía de Administración de Redes con Linux". *O'Reilly (printed version) (c) 2000 O'Reilly & Associates. Proyecto LuCAS por la traducción al español*. <http://es.tldp.org/Manuales-LuCAS/GARL2/garl2/>
- Red Hat, Inc (2005) "Red Hat Enterprise Linux 4 - Introducción a la administración de sistemas" <http://www.redhat.com/docs/manuals/enterprise/RHEL-4-Manual/es/admin-guide/>
- Red Hat, Inc (2005) "Red Hat Enterprise Linux 4 - Manual de Referencia" <http://www.redhat.com/docs/manuals/enterprise/RHEL-4-Manual/es/ref-guide/>

UNIDAD V

ADMINISTRACIÓN DEL SISTEMA GNU/Linux

La unidad tiene como propósito que el estudiante comprenda el funcionamiento de los recursos del sistema operativo GNU/Linux, valorando la importancia de los conocimientos para su desarrollo académico. Comprende:

- Supervisión de Recursos

Lección 10

Supervisión de Recursos

10.1. Comando df

Provee información sobre la utilización del espacio en disco en los diferentes sistemas de archivos montados en el sistema.

Sintaxis: df [opciones] [sistema-de-archivo...]

Si no se provee del argumento sistema-de-archivo, df informará acerca de todos los sistemas de archivos montados y en funcionamiento.

Las opciones de df más relevantes son:

- h Imprimir los tamaños de forma más legible.
- i Informar sobre la utilización de los nodos-í. Los nodos-í son estructuras internas del sistema de archivos, cuando éste se queda sin nodos-í libres, por mas que haya espacio libre en disco, no se podrán crear nuevos archivos hasta que se liberen nodos-í, generalmente esto no pasa a menos que se generen una enorme cantidad de archivos muy pequeños.
- k Mostrar los tamaños en bloques de 1024 bytes.
- m Mostrar los tamaños en bloques de mega-bytes.

Ejemplo:

```
[root@fisct ~]# df
S.ficheros Bloques de 1K      Usado      Dispon   Uso% Montado en
/dev/hda1    7656216          2933600    4327420  41%   /
Tmpfs       257316           0          257316   0%   /dev/shm
```

10.2. Comando du

El comando du informa de la cantidad de espacio de disco usada por los ficheros especificados, y por cada directorio en las jerarquías cuyas raíces estén en los ficheros especificados.

Sintaxis: du [opciones] archivo_o_ruta

Sus opciones más comunes son:

- -a: Muestra números para todos los ficheros, no sólo directorios.
- -b: muestra los tamaños en bytes.
- -k: muestra los tamaños en kilobytes.
- -h: un poco más amigable. Añade letra del tamaño.

Ejemplo:

```
[root@fisct ~]# du -sh /boot/grub/*
8,0K  /boot/grub/device.map
12K   /boot/grub/e2fs_stage1_5
12K   /boot/grub/fat_stage1_5
12K   /boot/grub/ffs_stage1_5
4,0K   /boot/grub/grub.conf
```

10.3. Comando ps

El comando **ps** nos permite ver los procesos que actualmente se están ejecutando en el sistema. Es un comando con una amplia parametrización para que podamos ver la información de procesos. Así mismo, nos permite visualizar las características de los procesos.

Sintaxis: ps [opciones]

Ejemplo:

```
[root@fisct ~]# ps
```

PID	TTY	TIME	CMD
3765	pts/1	00:00:00	bash
3808	pts/1	00:00:00	ps

PID	: Identificador del proceso (process identifier).
TTY	: Terminal
TIME	: Tiempo que ha usado (o usa) el proceso.
CMD (COMMAND)	: Nombre del proceso.

Listar todos los procesos de nuestra máquina:

```
[root@fisct ~]# ps -ef
```

root	1	0	0	Jan19	?	00:00:00	init [5]
root	2	1	0	Jan19	?	00:00:00	[migration/0]
root	3	1	0	Jan19	?	00:00:00	[ksoftirqd/0]
root	4	1	0	Jan19	?	00:00:00	[watchdog/0]
root	5	1	0	Jan19	?	00:00:00	[events/0]
root	6	1	0	Jan19	?	00:00:00	[khelper]
root	7	1	0	Jan19	?	00:00:00	[kthread]
root	10	7	0	Jan19	?	00:00:00	[kblockd/0]
root	11	7	0	Jan19	?	00:00:00	[kacpid]
root	98	7	0	Jan19	?	00:00:00	[cqueue/0]
root	101	7	0	Jan19	?	00:00:00	[khubd]
root	103	7	0	Jan19	?	00:00:00	[kseriod]
root	162	7	0	Jan19	?	00:00:00	[pdfflush]
root	163	7	0	Jan19	?	00:00:00	[pdfflush]
root	164	7	0	Jan19	?	00:00:00	[kswapd0]
root	165	7	0	Jan19	?	00:00:00	[aio/0]
root	321	7	0	Jan19	?	00:00:00	[kpsmoused]
root	344	7	0	Jan19	?	00:00:00	[ata/0]
root	345	7	0	Jan19	?	00:00:00	[ata_aux]
root	352	7	0	Jan19	?	00:00:00	[kstriped]
root	361	7	0	Jan19	?	00:00:00	[kjournald]
root	387	7	0	Jan19	?	00:00:00	[kauditfd]
root	420	1	0	Jan19	?	00:00:00	/sbin/udevd -d
root	1421	7	0	Jan19	?	00:00:00	[kmpathd/0]
root	1422	7	0	Jan19	?	00:00:00	[kmpath_handlerd]
root	1725	1	0	Jan19	?	00:00:00	auditd
root	1727	1725	0	Jan19	?	00:00:00	/sbin/audispd
root	1749	1	0	Jan19	?	00:00:00	syslogd -m 0
root	1752	1	0	Jan19	?	00:00:00	klogd -x
.....							
.....							

Resumen

En esta última unidad, se describe los comandos básicos para supervisar y gestionar los recursos del sistema.

Lectura

Proceso de arranque, inicio y cierre del sistema

Entre las características más importantes de GNU/Linux es el método abierto y configurable para el inicio y cierre del sistema operativo. Los usuarios son libres de configurar muchos aspectos en el proceso de arranque, incluyendo qué programas iniciarán al momento de su arranque. De forma parecida, el cierre del sistema finaliza los procesos de forma organizada y configurable, aunque la personalización de este proceso casi nunca es necesaria.

Entender el funcionamiento del proceso de arranque y cierre no sólo le permitirá personalizar, sino que también le facilitará resolver problemas relacionados con el inicio y el cierre del sistema.

1. Proceso de arranque

Entre las etapas básicas del proceso de arranque para un sistema x86 es el siguiente:

1. La BIOS (Basic Input-Output System) del sistema comprueba y lanza la primera etapa del gestor de arranque del MBR (master boot record) del disco duro.
2. La primera etapa del gestor de arranque se autocarga en memoria y lanza la segunda etapa del gestor de arranque desde la partición /boot/.
3. La segunda etapa del gestor de arranque carga el kernel en memoria, lo cual en su momento carga los módulos necesarios y monta la partición root para sólo-lectura.
4. El kernel transfiere el control del proceso de arranque al programa /sbin/init.
5. El programa /sbin/init carga todos los servicios y herramientas de espacio del usuario y monta todas las particiones listadas en /etc/fstab.
6. Se le presenta al usuario una pantalla de inicio de conexión para ingresar al sistema.

2. Descripción del proceso de arranque

El inicio del proceso de arranque varía dependiendo de la plataforma de hardware usada. Sin embargo, una vez que se encuentra el kernel y se carga por el gestor de arranque, el proceso de arranque por defecto es idéntico a través de todas las arquitecturas.

2.1. La BIOS (Basic Input/Output System)

Cuando un ordenador x86 se carga, el procesador busca al final de la memoria del sistema por Basic Input/Output System o programa BIOS y lo ejecuta. La BIOS controla no sólo el primer paso del proceso de arranque, sino que también proporciona una interfaz de bajo nivel para dispositivos periféricos. Por este motivo se escribe tan sólo en modo lectura, memoria permanente y está siempre disponible para el uso.

Una vez que se haya cargado, la BIOS chequea los periféricos y localiza un dispositivo para arrancar el sistema. En primer lugar comprueba cualquier dispositivo de entrada y/o unidades de CD-ROM presente por los medios de arranque, y a continuación si esto falla, verifica las unidades de disco duro del sistema. En la mayoría de los casos, el orden de búsqueda de las unidades para arrancar es controlado por una configuración de la BIOS y busca por el dispositivo maestro IDE en el bus IDE primario. La BIOS carga en memoria cualquier programa que resida en el primer sector de este dispositivo, llamado Registro de arranque principal o Master Boot

Record (MBR). La MBR sólo tiene 512 bytes de tamaño y contiene las instrucciones de código de máquina para el arranque del equipo, llamado gestor de arranque, así como también la tabla de particiones. Una vez que la BIOS haya encontrado y cargado el gestor de arranque en memoria, le deja el control del proceso de arranque a éste.

2.2. El gestor de arranque

Un gestor de arranque para la plataforma x86 se divide en al menos dos etapas. La primera es un código binario de máquina pequeña en el MBR. Su única función es la de localizar el gestor de arranque de la segunda etapa y cargar la primera parte de éste en memoria.

GRUB tiene la ventaja de ser capaz de leer particiones ext2 y ext3, cargar su archivo de configuración — /boot/grub/grub.conf — al momento del arranque.

Una vez que la segunda etapa del gestor de arranque está en memoria, presenta al usuario una pantalla gráfica mostrando los diferentes sistemas operativos o kernels que para los que ha sido configurado para arrancar. En esta pantalla el usuario puede usar las flechas direccionales para escoger el sistema operativo o kernel con el que desea arrancar y presione la tecla [ENTER]. Si no se presiona ninguna tecla, el gestor de arranque carga la selección predeterminada luego de un período de tiempo de espera.

Una vez que el gestor de arranque de la segunda etapa haya determinado qué kernel arrancar, localizará el binario del kernel correspondiente en el directorio /boot/. El kernel binario es llamado usando el siguiente formato — /boot/vmlinuz-<kernel-version> (donde <kernel-version> corresponde a la versión del kernel especificada en las configuraciones del gestor de arranque).

El gestor de arranque luego coloca una o más de las imágenes apropiadas de initramfs en la memoria. Luego, el kernel descomprime estas imágenes desde la memoria a /boot/, un sistema de archivos virtual basado en RAM, a través de cpio. El initrd es usado por el kernel para cargar controladores y módulos necesarios para arrancar el sistema. Esto es muy importante si posee unidades de disco duro SCSI o si el sistema utiliza el sistema de archivos ext3.

Una vez que el kernel y la imagen initramfs se cargan en memoria, el gestor de arranque pasa el control del proceso de arranque al kernel.

2.3. El kernel

Cuando se carga el kernel, éste inicializa y configura la memoria del ordenador y el hardware conectado al sistema, incluyendo todos los procesadores, subsistemas de entrada/salida y dispositivos de almacenamiento. A continuación buscará la imagen comprimida de initramfs en una ubicación predeterminada en memoria, la descomprime directamente a /sysroot/ y carga todos los controladores necesarios. A continuación inicializa los dispositivos virtuales relacionados con el sistema de ficheros, tales como LVM (Logical Volume Manager) o software RAID (Redundant Array of Independent Disks) antes de completar los procesos initramfs y de liberar toda la memoria que la imagen del disco ocupó anteriormente.

El kernel luego crea un dispositivo root, monta la partición root como sólo lectura y libera cualquier memoria no utilizada.

Llegados a este punto, el kernel estará cargado en memoria y operativo. Sin embargo, como no hay aplicaciones de usuario que permitan la entrada de datos al sistema, no se puede hacer mucho más.

Para configurar el entorno de usuario, el kernel inicia el programa /sbin/init.

2.4. Programa /sbin/init

El programa /sbin/init (también llamado init) coordina el resto del proceso de arranque y configura el ambiente del usuario.

Cuando el comando init arranca, se vuelve el padre o abuelo de todos los procesos que comienzan automáticamente en el sistema. Primero, ejecuta el script /etc/rc.d/rc.sysinit, que establece la ruta del entorno, activa el swap, verifica los sistemas de archivos y se encarga de todo lo que el sistema necesita tener al momento de la inicialización. Por ejemplo, la mayoría de los sistemas usan un reloj, por lo tanto, el rc.sysinit lee el archivo de configuración para iniciar el hardware del reloj. Otro ejemplo es con la configuración del hostname del ordenador, rc.sysinit ejecutará el archivo /etc/sysconfig/network.

El comando init luego ejecuta el script /etc/inittab, el cual describe cómo se debería configurar el sistema en cada nivel de ejecución SysV init. Los niveles de ejecución son un estado, o modo, definido por los servicios listados en el SysV directorio /etc/rc.d/rc<x>.d/, donde <x> es el número de nivel de ejecución.

Luego, el comando init configura la biblioteca de funciones fuente, /etc/rc.d/init.d/functions, para el sistema, que establece el modo en cómo iniciar o matar un programa y cómo determinar el PID del programa.

El programa init inicia todos los procesos de fondo buscando en el directorio apropiado rc para el nivel de ejecución especificado por defecto en /etc/inittab. Los directorios rc están numerados para corresponder al nivel de ejecución que representan. Por ejemplo, /etc/rc.d/rc5.d/ es el directorio para el nivel de ejecución 5.

Cuando se arranca el nivel de ejecución 5, el programa init consulta el directorio /etc/rc.d/rc5.d/ para determinar qué procesos iniciar o parar.

A continuación un ejemplo de listado del directorio /etc/rc.d/rc5.d/:

```
K05innd -> ../init.d/innd
K05saslauthd -> ../init.d/saslauthd
K12mailman -> ../init.d/mailman
K12mysqld -> ../init.d/mysqld
K15httpd -> ../init.d/httpd
K25squid -> ../init.d/squid
K30spamassassin -> ../init.d/spamassassin
K34dhcrelay -> ../init.d/dhcrelay
K34yppasswdd -> ../init.d/yppasswdd
K35dhcpd -> ../init.d/dhcpd
K35smb -> ../init.d/smb
K35vncserver -> ../init.d/vncserver
K50vsftpd -> ../init.d/vsftpd
K54dovecot -> ../init.d/dovecot
K61ldap -> ../init.d/ldap
S08iptables -> ../init.d/iptables
S10network -> ../init.d/network
S40smartd -> ../init.d/smard
S75postgresql -> ../init.d/postgresql
S80sendmail -> ../init.d/sendmail
```

Del ejemplo anterior, ninguno de los scripts que inician y cierran los servicios están localizados en el directorio /etc/rc.d/rc5.d/. Casi todos los ficheros en /etc/rc.d/rc5.d/ son enlaces simbólicos apuntando a los scripts localizados en el directorio /etc/rc.d/init.d/. Los enlaces simbólicos se usan en cada uno de los directorios rc de manera que los niveles de ejecución puedan ser reconfigurados al crear, modificar y eliminar los enlaces simbólicos sin que afecte a los scripts actuales a los que se refiere.

El nombre de cada enlace simbólico comienza con K o S. Los enlaces K son procesos eliminados en ese nivel de ejecución, mientras que aquellos que inician por S son procesos a iniciar.

El comando init en primer lugar detiene todos los enlaces simbólicos de K en el directorio mediante la ejecución del comando /etc/rc.d/init.d/<command> stop, en el que <command> es el proceso a matar. A continuación inicia todos los enlaces simbólicos S al ejecutar /etc/rc.d/init.d/<command> start.

Cada uno de los enlaces simbólicos se numera para dictaminar el orden de inicio. Se puede cambiar el orden en el que los servicios inician o paran al cambiar este número. Mientras más bajo es el número, más rápido se arrancará. Los enlaces simbólicos con el mismo número se inician de modo alfabético.

Una de las últimas cosas que el programa init ejecuta es el archivo /etc/rc.d/rc.local. Este archivo es útil para la personalización del sistema.

Después que el comando init ha progresado a través del directorio adecuado rc para el nivel de ejecución, el script /etc/inittab bifurca un proceso llamado /sbin/mingetty para cada consola virtual (prompt de inicio de sesión) del nivel de ejecución. Los niveles de ejecución del 2 al 5 tienen seis consolas virtuales, mientras que el nivel de ejecución 1 (modo usuario único) tiene tan sólo uno y los niveles de ejecución del 0 al 6 no tienen ninguno. El proceso /sbin/mingetty abre las rutas de la comunicación para los dispositivos tty, establece sus modos, imprime el indicador de inicio de sesión, toma el nombre y contraseña del usuario e inicia el proceso de inicio de sesión.

En el nivel de ejecución 5, /etc/inittab ejecuta un script llamado /etc/X11/prefdm. El script pref dm ejecuta su gestor de pantalla de X preferido — gdm, kdm, o xdm, dependiendo de los contenidos del archivo /etc/sysconfig/desktop.

Una vez que haya terminado, el sistema operará en el nivel de ejecución 5 y mostrará la pantalla de inicio de sesión.

3. Ejecutar programas adicionales en el momento de arranque

El script /etc/rc.d/rc.local lo ejecuta el comando init en tiempo de arranque, o cuando se cambien niveles de ejecución. Agregar comandos al final de este script es una forma fácil de realizar tareas necesarias como arrancar servicios especiales o inicializar dispositivos sin tener que escribir scripts complejos de inicialización en el directorio /etc/rc.d/init.d/ y creando enlaces simbólicos.

Ejemplo:

```
[root@fisct ~]# more /etc/rc.local  
#!/bin/sh  
#  
# This script will be executed *after* all the other init scripts.
```

```

# You can put your own initialization stuff in here if you don't
# want to do the full Sys V style init stuff.

/usr/local/apache/bin/apachectl start
/usr/sbin/postfix start
/usr/lib/courier-imap/libexec/imapd.rc start

```

Del ejemplo anterior muestra los servicios de apache (Servidor de Correo), postfix (Servidor de Correo) e imap (Servicio de Correo Entrante) han sido configurados para que sus servicios se inicien a partir del rc.local.

4. Niveles de ejecución de SysV Init

El sistema de niveles de ejecución SysV init provee de un proceso estándar para controlar cuáles programas init lanza o detiene cuando se inicializa un nivel de ejecución. SysV init es más fácil de usar y más flexible que el proceso tradicional init estilo BSD (Berkeley Software Distribution).

Los ficheros de configuración para SysV init están en el directorio /etc/rc.d/. Dentro de este directorio, se encuentran los scripts rc, rc.local, rc.sysinit, y, opcionalmente, los scripts rc.serial así como los siguientes directorios:

```

init.d/
rc0.d/
rc1.d/
rc2.d/
rc3.d/
rc4.d/
rc5.d/
rc6.d/

```

El directorio init.d contiene los scripts usados por el comando /sbin/init cuando se controlan los servicios. Cada uno de los directorios numerados representa los seis niveles de ejecución predeterminados y configurados por defecto en centOS.

4.1. Niveles de ejecución

Detrás de los niveles de ejecución, de SysV init, gira alrededor del hecho que sistemas diferentes se pueden usar de formas diferentes. Por ejemplo, un servidor corre de forma más eficiente sin el consumo de recursos del sistema excesivo creado por el sistema X. Otras veces, el administrador del sistema puede necesitar operar el sistema en un nivel más bajo de ejecución para realizar tareas de diagnóstico, como reparar corrupción del disco duro en el nivel de ejecución 1.

Las características de un nivel de ejecución, determinan qué servicios son detenidos o iniciados por init. Por ejemplo, el nivel de ejecución 1 (modo usuario único) detiene cualquier servicio de red, mientras que el nivel 3 arranca estos servicios. Asignando servicios específicos a ser detenidos o iniciados en un nivel dado, init puede fácilmente cambiar el modo de la máquina sin que el usuario tenga que manualmente arrancar o detener servicios.

Los siguientes niveles de ejecución están definidos de forma predeterminada son:

Nivel	Servicio	Descripción
0	Halt	Este nivel detiene el sistema
1	Single User	Modo de administración. El sistema crea un shell con los privilegios del superusuario sin solicitar nombre de usuario o contraseña.
2	Multiuser	Modo de funcionamiento normal sin algunos servicios de red.
3	Multiuser + network	Como el modo 2 pero con todos los servicios de red activos, NFS por ejemplo.
4		Generalmente no utilizado
5	Modo gráfico multiusuario completo	Con una pantalla de inicio de sesión basada en X
6	Reboot	Se reinicia el sistema.
s,S	Emergency single user	Igual al nivel 1 pero sin acceder a los ficheros de configuración de inicio.

Generalmente, los usuarios utilizan el nivel de ejecución 3 o nivel de ejecución 5 — ambos modos multiusuario. Ya que los niveles de ejecución 2 y 4 no son usados, los usuarios a veces personalizan estos niveles para cubrir necesidades específicas.

El nivel de ejecución por defecto para el sistema está listado en /etc/inittab. Para saber el nivel de ejecución por defecto de un sistema, verifique en el archivo /etc/inittab la siguiente línea:

id:5:initdefault:

El nivel de ejecución predeterminado en este ejemplo es cinco, como indica el número después del punto y coma. Para cambiarlo, modifique /etc/inittab como usuario root.

Es posible cambiar al nivel de ejecución por defecto al momento del arranque modificando los argumentos pasados por el gestor de arranque al kernel.

4.2. Utilidades de los niveles de ejecución

Para configurar los niveles de ejecución es usando la utilidad initscript. Estas herramientas están diseñadas para simplificar las tareas de mantener archivos en la jerarquía del directorio SysV init y permitir a los administradores de sistemas de tener que directamente manipular numerosos enlaces simbólicos en los subdirectorios de /etc/rc.d/.

CentOS ofrece tres utilidades:

- **/sbin/chkconfig** — La utilidad /sbin/chkconfig es una herramienta de línea de comandos sencilla para mantener la jerarquía del directorio /etc/rc.d/init.d.

Sintaxis:

```
chkconfig --list [name] chkconfig [--level levels] name <on|off|reset>
```

- **/sbin/ntsysv** — La utilidad basada en ncurses **/sbin/ntsysv** provee de una interfaz interactiva basada en texto, que muchos encuentran más fácil de usar que chkconfig.

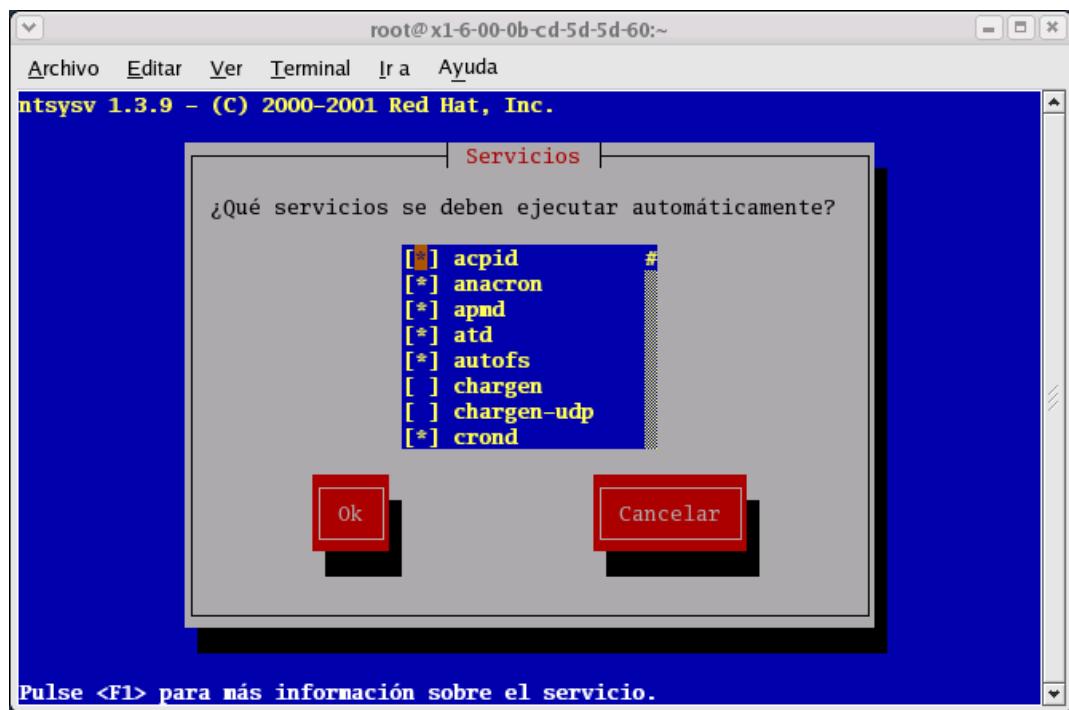


Figura1. Herramienta ntsysv

- **Herramienta de configuración de servicios** — El programa de interfaz gráfica **Herramienta de configuración de servicios** (system-config-services) es una utilidad flexible para la configuración de niveles de ejecución.

Fuente [2]

Glosario

- *Software Libre*: se refiere a la libertad de los usuarios para ejecutar, copiar, distribuir, estudiar, cambiar y mejorar el software. De modo más preciso, se refiere a cuatro libertades de los usuarios del software:
 - La libertad de usar el programa, con cualquier propósito (libertad 0).
 - La libertad de estudiar cómo funciona el programa, y adaptarlo a tus necesidades (libertad 1). El acceso al código fuente es una condición previa para esto.
 - La libertad de distribuir copias, con lo que puedes ayudar a tu vecino (libertad 2).
 - La libertad de mejorar el programa y hacer públicas las mejoras a los demás, de modo que toda la comunidad se beneficie. (libertad 3). El acceso al código fuente es un requisito previo para esto.
- *System V*: abreviado como SysV ó denominado System 5, fue una de las versiones del sistema operativo Unix.
- *BSD*: (Berkeley Software Distribution). BSD es un sistema operativo derivado de Unix, distribuido por la Universidad de California desde los 70.
- *FTP*: (File Transfer Protocol - Protocolo de Transferencia de Archivos). Es un sistema que permite enviar y recibir ficheros entre computadores a través de la red Internet.
- *rpm*: (Redhat Package Manager) es una herramienta utilizada para instalar, actualizar, desinstalar, verificar y solicitar programas cuya extensión se reconoce como .rpm
- *yum*: es una herramienta que permite instalar/desintalar paquetes rpm desde un servidor remoto o llamado también como mirrors.
- *grub*: (GRand Unified Bootloader) es un gestor de arranque múltiple utilizado para iniciar dos más sistemas operativos instalado en un mismo ordenador.
- *Gnome*: Es una interfaz de escritorio donde el usuario podrá ejecutar programa, manejas ficheros y administrar ventanas.
- *KDE*: Es un interfaz de escritorio similar al GNOME.
- *NFS*: (Network File System) es un sistema de archivos virtual que permite que una máquina UNIX, conectada a una red, pueda montar un sistema de archivos de otra máquina e interactuar sobre él como si fuera propio.

Autoevaluación

Marcar la respuesta correcta:

1. Es el encargado de controlar no sólo el primer paso del proceso de arranque, sino que también proporciona una interfaz de bajo nivel para dispositivos periféricos. Además de localizar el dispositivo con el que arrancará el sistema.

- a. shell ()
- b. Gestor de arranque ()
- c. BIOS ()
- d. kernel ()
- e. Niveles de Ejecución ()

2. Considerado como el proceso padre de todos los procesos que se cargan automáticamente en el sistema.

- a. kernel ()
- b. grub ()
- c. initrd ()
- d. MBR ()
- e. init ()

3. Comando correspondiente al nivel de ejecución 6.

- a. poweroff ()
- b. reboot ()
- c. halt ()
- d. shutdown -h now ()
- e. exit ()

4. Cuando el sistema carga el nivel de ejecución.... lanza una interfaz en modo texto.

- a. 0 ()
- b. 1 ()
- c. 3 ()
- d. 5 ()
- e. 6 ()

5. Es un ejemplo de proceso a matar o dar baja en el sistema:

- a. /sbin/init ()
- b. K28amd ()
- c. startx ()
- d. /etc/rc.d/ ()
- e. S75postgresql ()

6. No es considerado como nivel de ejecución multiusuario:

- a. 2 ()
- b. 3 ()
- c. 4 ()
- d. 5 ()
- e. 6 ()

7. Es el encargado de abrir las sesiones de comunicación. Establece sus modos, muestra el indicador de inicio de sesión, ingresando el nombre y contraseña del usuario permitiendo el inicio de sesión.

- a. /boot/grub/grub.conf ()
- b. /etc/inittab ()
- c. /sbin/mingetty ()
- d. /sbin/init ()
- e. /etc/fstab ()

8. Se encarga de montar las particiones (/boot, swap y /):

- a. /boot/grub/grub.conf ()
- b. /etc/inittab ()
- c. /sbin/mingetty ()
- d. /sbin/init ()
- e. /etc/fstab ()

9. Comando correspondiente al nivel de ejecución 0:

- a. exit ()
- b. reboot ()
- c. halt ()
- d. logout ()
- e. Todas la Anteriores ()

10. Es un script que permite agregar líneas de comandos. Se utiliza para arrancar servicios especiales:

- a. /etc/init.d/xinetd ()
- b. /boot/grub/grub.conf ()
- c. /etc/rc.d/rc.local ()
- d. /etc/inittab ()
- e. /etc/sysconfig/network ()

Solucionario

Marcar la respuesta correcta:

1. c
2. e
3. b
4. c
5. b
6. e
7. c
8. e
9. c
10. c

Bibliografía

- [1]. Negus, Christopher (2003) *Red Hat Linux 8*, Madrid. Ediciones Anaya Multimedia.
- [2]. Red Hat, Inc (2005) "Red Hat Enterprise Linux 4 - Manual de Referencia"
<http://www.redhat.com/docs/manuals/enterprise/RHEL-4-Manual/es/ref-guide/>

Enlaces

- Kirch, Olaf y Dawson, Ferry (2002) "Guía de Administración de Redes con Linux".
O'Reilly (printed version) (c) 2000 O'Reilly & Associates. Proyecto LuCAS por la traducción al español.
<http://es.tldp.org/Manuales-LuCAS/GARL2/garl2/>
- Red Hat, Inc (2005) "Red Hat Enterprise Linux 4 - Introducción a la administración de sistemas"
<http://www.redhat.com/docs/manuals/enterprise/RHEL-4-Manual/es/admin-guide/>