GNU/Linux, software libre para la comunidad universitaria

Administración avanzada del sistema

COLABORADORES

	TÍTULO :	*REFERENCE* :
	GNU/Linux, software libre para la comunidad universitaria	

ACCIÓN	*NOMBRE*	*FECHA*	*FIRMA*
ESCRITO POR	Pablo Cabezas Mateos y José Ángel de Bustos Pérez	22 de abril de 2008	

HISTORIAL DE REVISIONES

NÚMERO	FECHA	MODIFICACIONES	NOMBRE
1.0	10-04-2007		Pablo Cabezas Mateos, José Ángel de Bustos Pérez
1.1	16-04-2008		Pablo Cabezas Mateos, José Ángel de Bustos Pérez

Índice general

Índice de cuadros

Capítulo 1

Introducción

Este manual se ha realizado como parte de la documentación del curso de que imparte la Universidad de Salamanca, Augcyl y GLiSA: *Utilización y Administración Avanzadas de Sistemas GNU/Linux y aplicaciones Software Libre para estudiantes Universitarios*.

Una de las ponencias de curso es la Administración Avanzada del Sistema GNU/Linux, la cual se plasma en este manual. En esta ponencia se presenta todas las posibilidades de la administración del sistema, y en el resto de ponencias entrarán en detalle de algunos de los conceptos más importantes de administración que en esta documentación encontramos.

1.1. Objetivo

En este manual vamos a intentar dar las características generales que tiene un sistema GNU/Linux y que la persona encargada, el administrador, debe conocer para "mantener" el sistema.

Un sistema GNU/Linux nos lo encontramos como un conjunto de aplicaciones reunidas entorno a núcleo del sistema, es lo que denominamos distribución. No vamos a entrar en las particularidades de cada distribución, aunque el curso se imparte con la distribución SuSE. Vamos a intentar ver las generalidades de todas ellas, que a su vez son parecidas a los sistemas operativos UNIX, del cual Linux hereda.

Tampoco intenta ser un manual de sistema operativo (SO) UNIX, pero si vamos a dividir los capítulos por componentes de un Sistema Operativo.

Con la ponencia y con este manual de apoyo queremos lograr que el alumno obtenga los conceptos básicos de la administración de GNU/Linux, las responsabilidades del administrador de GNU/Linux y pueda afrontar cualquier reto que la administración de GNU/Linux pueda plantearles.

Capítulo 2

Gestión de Procesos

La ejecución de programas en Linux se realiza mediante procesos que se están ejecutando a la vez. El núcleo de sistema operativo realiza una gestión para determinar que proceso debe ejecutarse en los procesador/es. El administrador debe saber que es lo que se esta ejecutando y como puede optimizar sus procesos.

Los procesos no se ejecutan completamente cuando se arrancan, el núcleo ejecuta el proceso durante un pequeño tiempo, dando la sensación que se ejecutan varios procesos a la vez, esto se denomina multitarea.

2.1. Estados de procesos

Los procesos pueden estar en distintas situaciones según el momento en que esta el sistema o el programa. Vamos ver los estados que existen.

2.1.1. RUN

Indica que el proceso esta ejecutándose en la CPU en ese momento.

2.1.2. READY

Cuando el estado de un proceso es READY dicho proceso está preparado para ser ejecutado, pero la CPU está ejecutando otro proceso por lo que esta a la espera de que quede libre para comenzar a ejecutarse.

2.1.3. WAIT

El proceso está a la espera de obtener un recurso del sistema.

2.1.4. STOPPED

El proceso está parado. Mediante una señal se le deja en un estado que no realiza ningún tipo de ejecución.

2.1.5. ZOMBIE

Es un estado intermedio antes de desaparecer del sistema una vez que ha terminado el proceso. Pueden quedarse en estado procesos que han terminado correctamente y otros procesos tienen referencias a este proceso.

2.2. Atributos de los procesos

Vamos a ver los atributos más importantes de los procesos.

2.2.1. Identificación de proceso (PID)

A cada proceso el sistema le asigna un número que lo identifica unívocamente. Va a ser el atributo con el cual realicemos operaciones sobre el proceso.

2.2.2. Identificación del proceso padre (PPID)

Todo proceso es arrancado por otro proceso que se le denomina padre. Su PID queda en el hijo para que pueda formarse el árbol de procesos.

Cuando arrancamos la máquina el sistema crea al proceso init que le da el PID 1, del que parten el resto de procesos.

2.2.3. Usuario (UID)

Es el usuario que ejecutan el proceso.

2.2.4. Grupo (GID)

El grupo del usuario que lanzo del proceso.

2.2.5. Prioridad

Para la planificación de ejecución que realiza el sistema los procesos llevan un valor que es la prioridad. Este valor determinará el tiempo que le dedicará a ese proceso.

Tanto el usuario propietario del proceso como root puede modificar el valor de la prioridad.

2.3. Comandos de gestión de procesos

Vamos ver los comandos con los que podemos ver el estado y atributos de los procesos y los comandos que nos permiten cambiar algunas características de los procesos.

2.3.1. Ejecución en background y comandos jobs, fg y bg

Cuando ejecutamos un comando en la consola esta se queda a la espera que termine el proceso que hemos ejecutado. Podemos hacer que comience la ejecución e inmediatamente nos devuelva el control a la consola. Para ello tenemos que utilizar el operador & al final del comando.

```
[pcm@sal]$ firefox &
[1] 23710
[pcm@sal]$
```

Si no hubiésemos puesto & hasta que no cerráramos la aplicación firefox no podríamos usar la consola. Podríamos ver que estamos ejecutando en esa consola en background con el comando **jobs**.

```
[pcm@sal]$ jobs
[1]+  Running                 firefox &
[pcm@sal]$
```

Incluso podemos llevar una aplicación que esta corriendo en background a la consola con el comando **fg** dándole el número que aparece con el comando jobs o cuando ejecutamos el comando en background. También podemos parar un proceso que esta en la consola pulsando Control+Z o utilizando desde otra consola el comando **kill**, y luego enviando el proceso a background con **bg**.

```
[pcm@sal]$fg 1
firefox
```

Si ahora queremos llevarle a background de nuevo pulsaríamos Control+Z.

```
[1]+  Stopped                 firefox
[pcm@sal]$ bg 1
[1]+ firefox &
[pcm@sal]$
```

importante
Si un proceso en background utiliza la consola para interactuar con el usuario, el proceso se queda en estado parado.

2.3.2. Comando ps

Este comando nos permite ver los procesos que actualmente existen en el sistema. Es un comando con una amplia parametrización para que podamos ver cómodamente la información de procesos.

Este es el comando preparado para ver todas las características de los procesos. Al ser un comando estándar el todos los UNIX soporta un montón de parámetros especificados en versiones anteriores del **ps**.

Para listar todos los procesos de la máquina haríamos:

```
[root@sal]# ps -ef

UID        PID  PPID  C STIME TTY          TIME CMD
root         1     0  0 Apr07 ?        00:00:01 init [2]
root         2     1  0 Apr07 ?        00:00:02 [keventd]
root         3     1  0 Apr07 ?        00:00:00 [ksoftirqd_CPU0]
root         4     1  0 Apr07 ?        00:00:00 [ksoftirqd_CPU1]
root         5     1  0 Apr07 ?        00:00:03 [kswapd]
root         6     1  0 Apr07 ?        00:00:00 [bdflush]
root         7     1  0 Apr07 ?        00:00:03 [kupdated]
root       409     1  0 Apr07 ?        00:00:00 [knodemgrd_0]
root       541     1  0 Apr07 ?        00:00:00 [khubd]
daemon     865     1  0 Apr07 ?        00:00:00 /sbin/portmap
root       955     1  0 Apr07 ?        00:00:03 /sbin/syslogd
root       958     1  0 Apr07 ?        00:00:00 /sbin/klogd
pcm20     1026     1  0 Apr07 ?        00:00:00 /usr/sbin/famd -T 0
root      1032     1  0 Apr07 ?        00:00:00 /usr/sbin/inetd
root      1047     1  0 Apr07 ?        00:00:00 /usr/sbin/lisa
root      1150     1  0 Apr07 ?        00:00:01 /usr/sbin/nmbd -D
root      1152     1  0 Apr07 ?        00:00:00 /usr/sbin/smbd -D
root      1158  1152  0 Apr07 ?        00:00:00 /usr/sbin/smbd -D
root      1159     1  0 Apr07 ?        00:00:00 /usr/sbin/sshd
root      1168     1  0 Apr07 ?        00:00:00 /sbin/rpc.statd
root      1174     1  0 Apr07 ?        00:00:01 /usr/sbin/sensord -f daemon
daemon    1205     1  0 Apr07 ?        00:00:00 /usr/sbin/atd
root      1208     1  0 Apr07 ?        00:00:00 /usr/sbin/cron
root      1313     1  0 Apr07 ?        00:00:02 /usr/sbin/apache
root      1325     1  0 Apr07 ?        00:00:00 /usr/bin/hts -F localhost:22
root      1341     1  0 Apr07 ?        00:00:00 /usr/bin/kdm
root      1348     1  0 Apr07 tty1     00:00:00 /sbin/getty 38400 tty1
```

```
root      1349     1  0 Apr07 tty2     00:00:00 /sbin/getty 38400 tty2
root      1350     1  0 Apr07 tty3     00:00:00 /sbin/getty 38400 tty3
root      1351     1  0 Apr07 tty4     00:00:00 /sbin/getty 38400 tty4
root      1352     1  0 Apr07 tty5     00:00:00 /sbin/getty 38400 tty5
root      1353     1  0 Apr07 tty6     00:00:00 /sbin/getty 38400 tty6
root      1372  1341  0 Apr07 ?        00:14:53 /usr/X11R6/bin/X -nolisten tcp -
root      1411     1  0 Apr07 ?        00:00:02 /usr/bin/perl /usr/share/webmin/
root      1418  1341  0 Apr07 ?        00:00:00 -:0
[root@sal]#
```

Es muy recomendable ver el manual de este comando.

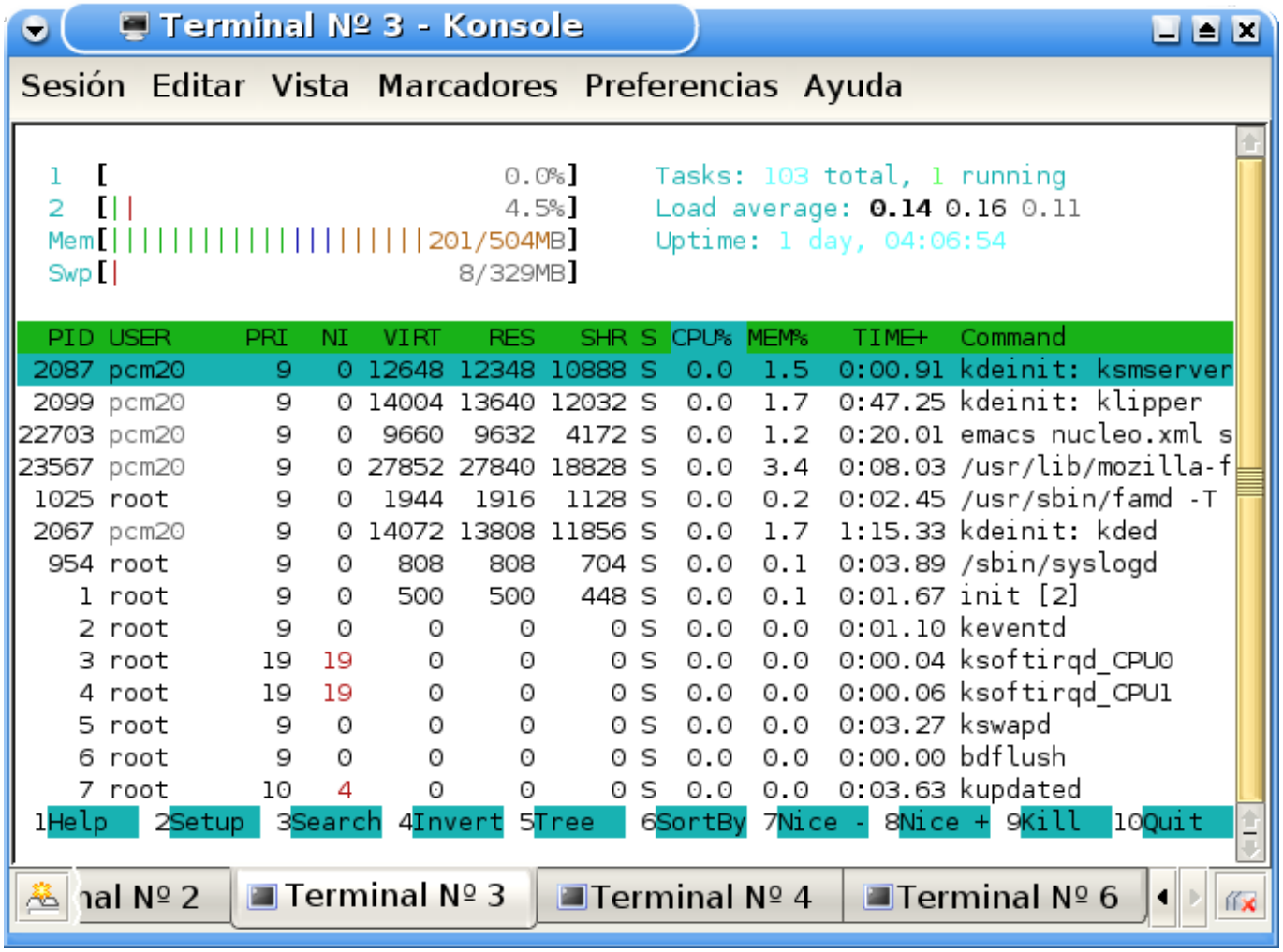
2.3.3. Comando top y htop

Nos muestra información de los procesos que existen, y de algunos parámetros del sistema como la memoria, tiempo de ocupación de cpu, a intervalos, es decir, no termina una vez que los muestra si no que vuelve a ejecutarse en pequeño tiempo y vuelve a mostrar los nuevos datos.

```
[root@sal]# top
top - 01:47:25 up 1 day,  3:38,  1 user,  load average: 0.01, 0.04, 0.03
Tasks: 112 total,   1 running, 111 sleeping,   0 stopped,   0 zombie
Cpu(s):   5.6% user,   8.2% system,   0.0% nice,  86.1% idle
Mem:    516528k total,   503012k used,    13516k free,    44320k buffers
Swap:   337356k total,    13496k used,   323860k free,   165572k cached

  PID USER      PR  NI  VIRT  RES  SHR S  %CPU  %MEM    TIME+  COMMAND
 2078 pcm       19   0 14772  14m  12m S 10.6  2.8  11:37.18 kdeinit
 1372 root      12 -10 83448  73m 9228 S  3.5 14.5  14:55.90 XFree86
22682 root      19   0  1084 1084  844 R  3.2  0.2   0:00.30 top
 2086 pcm       10   0  101m  97m  23m S  1.3 19.4  21:51.38 netscape-bin
21199 pcm       12   0 30420  29m  13m S  1.3  5.9   0:10.50 java_vm
 2100 pcm       12   0  101m  97m  23m S  1.0 19.4   0:09.66 netscape-bin
 2075 pcm       11   0 18184  17m  13m S  0.6  3.5   0:21.67 kdeinit
 5012 pcm       11   0 10864  10m 4072 S  0.6  2.0   3:13.78 emacs
 8591 mysql     10   0 13600  13m 2932 S  0.6  2.6   0:48.99 mysqld
 1411 root      10   0  5744 3448 2088 S  0.3  0.7   0:02.43 miniserv.pl
 2068 pcm        9   0 12920  12m  10m S  0.3  2.4   1:35.65 kdeinit
 8153 root       9   0   944  904  796 S  0.3  0.2   0:09.22 dirmngr
 8592 mysql     10   0 13600  13m 2932 S  0.3  2.6   0:54.86 mysqld
21197 pcm        9   0 30420  29m  13m S  0.3  5.9   0:04.70 java_vm
    1 root       9   0   500  500  448 S  0.0  0.1   0:01.62 init
    2 root       9   0     0    0    0 S  0.0  0.0   0:02.30 keventd
```

El mismo comando pero visualmente más bonito y más sencillo de utilizar es **htop**. Aunque es probable que no lo encontremos instalado como base en muchos sistema Linux o Unix.



2.3.4. Comando pstree

Nos da el árbol de procesos, es decir nos hace un árbol de los procesos mediante su PPID.

2.3.5. Comando nice y renice

Estos comandos nos permiten lanzar un proceso con una prioridad determinada el primero y el segundo cambiar la prioridad de un proceso.

2.3.6. Comando kill

Nos permite enviar una señal a un proceso. Por defecto envía la señal de parada al PID que pongamos, de ahí su nombre, pero realmente con un parámetro podemos enviar otros tipos de señales de procesos.

```
[root@sal]# kill 22670
[root@sal]#
```

Para ver la lista de señales soportadas por el comando podemos hacer:

```
[root@sal]# kill -l

 1) SIGHUP       2) SIGINT       3) SIGQUIT      4) SIGILL
 5) SIGTRAP      6) SIGABRT      7) SIGBUS       8) SIGFPE
```

```
 9) SIGKILL     10) SIGUSR1     11) SIGSEGV     12) SIGUSR2
13) SIGPIPE     14) SIGALRM     15) SIGTERM     17) SIGCHLD
18) SIGCONT     19) SIGSTOP     20) SIGTSTP     21) SIGTTIN
22) SIGTTOU     23) SIGURG      24) SIGXCPU     25) SIGXFSZ
26) SIGVTALRM   27) SIGPROF     28) SIGWINCH    29) SIGIO
30) SIGPWR      31) SIGSYS      33) SIGRTMIN    34) SIGRTMIN+1
35) SIGRTMIN+2  36) SIGRTMIN+3  37) SIGRTMIN+4  38) SIGRTMIN+5
39) SIGRTMIN+6  40) SIGRTMIN+7  41) SIGRTMIN+8  42) SIGRTMIN+9
43) SIGRTMIN+10 44) SIGRTMIN+11 45) SIGRTMIN+12 46) SIGRTMIN+13
47) SIGRTMIN+14 48) SIGRTMIN+15 49) SIGRTMAX-15 50) SIGRTMAX-14
51) SIGRTMAX-13 52) SIGRTMAX-12 53) SIGRTMAX-11 54) SIGRTMAX-10
55) SIGRTMAX-9  56) SIGRTMAX-8  57) SIGRTMAX-7  58) SIGRTMAX-6
59) SIGRTMAX-5  60) SIGRTMAX-4  61) SIGRTMAX-3  62) SIGRTMAX-2
63) SIGRTMAX-1  64) SIGRTMAX
[root@sal]#
```

Las señales más utilizadas son:

- SIGKILL Termina el proceso, no puede ser ignorada. Se puede enviar con `kill -s SIGKILL` *pid* o `kill -9` *pid*, que es modo estándar en otros UNIX.
- SIGSTOP Para el proceso, no puede ser ignorada.
- SIGTERM Se ordena al proceso que termine, el cual puede ser ignorar la señal. El la señal enviada por defecto.

2.3.7. Comando killall y pkill

Con estos comandos vamos a poder enviar señales a procesos, en vez de por PID, por el nombre, con **killall** y por otras propiedades de los procesos como pueden ser el usuario, el GID, el proceso padre, etc, con **pkill**. Por ejemplo:

```
[pcm@sal]# ps
 4548 pts/2    00:00:00 bash
 4565 pts/2    00:00:00 gconfd-2
22780 pts/2    00:00:34 xpdf.bin
22824 pts/2    00:00:00 ps

[pcm@sal]# killall xpdf.bin
[pcm@sal]#
```

Con esta acción terminaría el proceso 22780.

Capítulo 3

Gestión de Memoria

Linux es un sistema operativo con una gestión de memoria que esta preparada para memoria virtual y sistemas multiprocesador.

3.1. Memoria física

La memoria física es la memoria RAM que dispone el ordenador.

Esta memoria es el espacio que están utilizando los procesos que se están ejecutando. En Linux también se utiliza la memoria física para tener cache de los datos de los dispositivos de i/o, memoria compartida y buffers de intercambio.

Al ser un recurso caro, pero rápido, Linux va a intentar utilizar el máximo de ella, por ello cuando la memoria no es utilizada por las aplicaciones, es decir los procesos lanzados no ocupan toda la memoria, utiliza toda la que puede como cache de datos contra otros dispositivos más lentos como los discos duros. Cuando las aplicaciones vayan requiriendo más memoria estas caches serán más pequeñas.

La memoria está paginada y así poder llevar partes a la memoria virtual. También puede llevar procesos enteros.

3.2. Memoria virtual

Cuando algún espacio de memoria no es utilizado puede ser llevado a un dispositivo de almacenamiento. Con **free** podemos ver su utilización actual.

La memoria swap debemos indicar nosotros en donde queremos ubicarla. Para ello podemos dedicar una partición entera de nuestros discos duros o bien crear un fichero en uno de los sistemas de ficheros que lo soporten. Debemos tener en cuenta que es más óptimo la utilización de una partición pero para ello debemos tener el espacio reservado para la creación de la partición.

3.2.1. Creación de Swap

Para crear una partición de swap debemos utilizar el programa de particiones.

```
[root@sal]$ fdisk
Command (m for help): n
Command action
   e   extended
   p   primary partition (1-4)
```

Crearemos la partición del tipo y tamaño que deseemos y lo único que cambiaremos con la opción t es el id de la partición al tipo 82. Nos debe quedar algo así:

```
/dev/hda3           14552       14593      337365   82  Linux swap / Solaris
```

Si bien queremos crear un fichero de swap lo haremos con el siguiente comando.

```
[root@sal]$ mkswap /file-swap 1024
Setting up swapspace, size = 1044480
  bytes
```

Ya tenemos creada la swap ahora hay que indicar al sistema que la utilice.

3.2.2. Usando el Swap

Para empezar a utilizarla ejecutamos el comando swapon.

```
[root@sal]$ swapon /file-swap
```

Evidentemente para no estar haciendo esto cada vez que arrancamos se puede configurar en el fichero `/etc/fstab` que es donde están las particiones a montar (y ficheros swap) en el arranque.

Una vez puesta a disposición del sistema el gestor de memoria la utilizará cuando lo considere necesario.

3.3. Cache y Buffers

El comando **free** también nos da información de cuanta memoria se esta dedicando a caches, buffers y memoria compartida.

La cache es utiliza para guardar temporalmente información que esta en los dispositivos de I/O que pueden volver a necesitarse. Linux intentará utilizar el máximo de memoria física para este concepto ya que cuanto más tenga más posibilidades de que algo que se le pida de nuevo lo tenga ya en las caches. Automáticamente hará las caches más pequeñas si la memoria es necesitada por las aplicaciones.

3.4. Herramientas

Para ver el uso y análisis de la memoria el administrador dispone de algunas herramientas.

El rendimiento de la memoria virtual es muy importante ya que cuando vayamos a hacer uso de ella es porque al sistema se le esta requiriendo un esfuerzo mayor y no queremos que sea el cuello de botella. Por ello las herramientas nos deben dar las estadísticas de la utilización de la memoria.

3.4.1. Comando free

Para ver el estado de la memoria actual tenemos el comando **free**.

```
[pcm@sal]$ free -m
             total       used       free     shared    buffers     cached
Mem:           504        491         12          0         41        147
-/+ buffers/cache:        302        201
Swap:          329         51        277
[pcm@sal]$
```

La primera linea nos está dando como realmente se esta utilizando la memoria física, y la segunda cuanta memoria hay libre desde el punto de vista de las aplicaciones. La última linea contiene la utilización de la memoria swap.

En detalle, este sistema tiene 504Mb de memoria de los cuales están usados 491Mb y 12Mb libres. De los usados 302Mb por las aplicaciones, 41Mb de memoria compartida y 147Mb para buffers y caches. Como la memoria para buffers y cache es adaptable según las necesidades, tendríamos hasta 201Mb libres.

Muchas de las herramientas de administración nos dan también estos mismos valores de memoria como por ejemplo **top** o **htop** o desde webmin. Esta información se recoge del fichero `/proc/meminfo` del sistema de ficheros virtual del núcleo.

3.4.2. Comando vmstat

Nos da la misma información que **free** más algunos datos estadísticos sobre el uso de la memoria swap y entrada/salida con dispositivos de bloques.

Nos permite por parámetro lanzar la petición varias veces y cada cierto periodo de tiempo, así tendremos que poner **vmstat 2 10** para que se ejecute 10 veces y muestre los datos cada dos segundos.

```
[pcm@sal]$ vmstat 2 10
procs -----------memory---------- ---swap-- -----io---- --system-- ----cpu----
 r  b   swpd   free   buff  cache   si   so    bi    bo   in    cs us sy id wa
 0  0  66016 113748  19112 231988    0    0     2     3   38    23  1  1 98  0
 0  0  66016 113740  19112 231988    0    0     0     0  196   460  3  0 97  0
 1  0  66016 113844  19112 231988    0    0     0     0  193   427  5  1 94  0
 0  0  66016 113608  19112 232216    0    0   114     0  206   425  7  1 92  0
 0  0  66016 113524  19112 232248    0    0    16    12  206   422  6  1 93  0
 0  0  66016 113472  19112 232248    0    0     0     0  191   433  7  0 93  0
 0  0  66016 113472  19112 232252    0    0     2     0  190   384  5  1 94  0
 0  0  66016 113468  19112 232252    0    0     0     0  195   799  4  1 95  0
 2  0  66016 113468  19112 232252    0    0     0   192  227   727  8  1 91  0
 0  0  66016 113468  19112 232252    0    0     0     0  203   260  0  0 99  0
[pcm@sal]$
```

En cada linea nos va dando el estado de las variables. Serán tantas lineas como hayamos puesto en el segundo parámetro y van saliendo cada tantos segundo como hayamos puesto en el primero.

- *procs*: Nos da los procesos preparados para correr en una cpu y los procesos durmiendo.
- *memory*: Estado de la memoria. Los mismo datos que el comando **free**.
- *swap*: Nos da las páginas intercambiadas por segundo entre la memoria física y la memoria virtual.
- *io*: Bloques recibidos y enviados desde un dispositivo por segundo.
- *system*: Interrupciones lanzadas y cambio de contexto por segundo.
- *cpu*: Porcentajes de tiempos de cpu dedicados a sistema, a aplicaciones de usuario, sin hacer nada y tiempo dedicado a IO (Entrada/Salida).

3.4.3. Comando sar

Este comando permite realizar estadísticas de muchos más elementos, como red, cpu, dispositivos de bloques y memoria. Esta muy pensado para analizar el rendimiento de estos elementos. Para la memoria virtual dispone de muy diversas variables que nos pueden ayudar a ver donde puede haber un problema.

No vamos a entrar en detalle con todas las posibilidades de este comando.

Capítulo 4

Sistemas de Ficheros

Dentro del mundo UNIX todo son ficheros, debido a esta abstracción los sistemas UNIX tienen una gran potencia y versatilidad para el manejo de cualquier tipo de dispositivo.

No existe el concepto de dispositivo físico, por lo cual es habitual encontrarnos en una máquina UNIX con directorios que fisicamente estan en otro equipo en la red siendo transparente para el usuario.

4.1. Organización de directorios

El esquema de directorios y la organización de estos es bastante diferente en los sistemas UNIX del resto de sistemas.

Cada directorio tiene un cometido.

Cada sistema UNIX tiene una estructura ligeramente diferente al resto.

En los sistemas GNU/Linux pasa lo mismo lo cual no deja de ser un engorro. Para evitar esto se ha tratado de estandarizar la jeraquía del sistema de ficheros.

---

sugerencia
La referencia para esta estandarización la podemos encontrar en http://www.pathname.com/fhs/

4.2. Ficheros estándar

En los sistemas UNIX y en GNU/Linux en particular los ficheros estándar son:

- *stdinput* o Salida estándar.
- *stdoutput* o Entrada estándar.
- *stderr* o Salida estándar de errores.

4.2.1. La entrada estándar

Es el dispositivo que se utiliza por defecto para la entrada de datos.

Por defecto es el teclado y el descriptor de la entrada estándar es el *0*.

4.2.2. La salida estándar

Es el dispositivo que se utiliza por defecto, como su propio nombre indica, para mostrar la salida de datos.

Por defecto es el monitor y el descriptor de la salida estándar es el *1*.

4.2.3. La salida estándar de errores

Es el dispositivo que se utiliza por defecto para la salida de errores.

Por defecto es el monitor y el descriptor de la salida estándar de errores es el *2*.

4.3. Redirecciones

Aunque por defecto los ficheros estándar estan redirigidos a unos dispositivos en particular es posible cambiar esa redirección hacía otro dispositivo.

Para las redirecciones se utilizan los siguientes operadores:

- < se utiliza para para indicar el fichero del que se recogeran los datos en lugar de la entrada estándar.
- > se utiliza para redirigir hacía un fichero. El contenido del fichero es reemplazado.
- » se utiliza para añadir a un fichero. Se mantiene el contenido del fichero.
- *&n* donde *n* es un descriptor. Se utiliza para redirigir un flujo hacía donde ha sido redirigido un fichero.

4.3.1. Redirección de la salida estándar

Muchas veces es necesario recoger la salida de un comando en un fichero para su posterior proceso.

Habrá que tener en cuenta si la información que vamos a recoger se va a almacenar en un fichero con datos o en uno vacío. En caso de almacenarse en uno con datos tendremos que tener claro si vamos o no a necesitar los datos ya existentes. En función de esto utilizaremos el operador > o »:

```
[pcm@sal]$ ls -lh > contents.dir
```

La cual eliminaría el contenido del fichero `contents.dir` reemplazandolo con el listado del directorio. O bien:

```
[pcm@sal]$ ls -lh >> contents.dir
```

Que añadiría el listado de directorios al contenido del fichero `contents.dir`.

4.3.2. Redirección de la entrada estándar

Es posible sustituir la entrada de datos a través del teclado mediante el operador <.

Esto es útil cuando queremos lanzar scripts de forma automatizada sin la intervención del usuario y es necesario el suplir información como login o contraseña.

4.3.3. Redirección de la salida estándar de errores

Aunque por defecto tanto la salida estándar como la de errores estan ambas redirigidas hacía el monitor en realidad apuntan a dos ficheros diferentes.

Gracias a esto es posible separar ambas salidas para procesar por separado los errores y la salida del programa.

Para referirnos a la salida estándar de errores lo haremos mediante el operador *2>*:

```
[pcm@sal]$ ls -lh >> contents.dir 2> errores
```

Se añadiráa el listado de directorios al fichero `contents.dir` y los mensajes de errores que pudieran ocurrir en lugar de mostarse por la pantalla se almacenarán en el fichero `errores`.

importante
En realidad lo que estamos haciendo es utilizando el descriptor de la salida estándar e indicandole el tipo de redirección que queremos hacer.

4.3.4. El operador *&n*

Hay veces que es necesario redirigir tanto la salida como la salida estándar a un mismo fichero. Por ejemplo durante una compilación ya que es necesario conocer el orden de los mensajes, tanto de la compilación como de los errores.

Cuando queramos hacer una redirección hacía un fichero sobre el que ya se ha hecho una redirección utilizaremos el operador *&n* donde *n* es el descriptor del fichero que ha sido redirigido previamente a ese fichero:

```
[pcm@sal]$ make all > resultados 2>&1
```

En el ejemplo anterior estamos haciendo lo siguiente:

- Redirigimos hacía el fichero `resultados` la salida estándar del comando **make all**.
- Mediante *2>* indicamos que queremos redirigir la salida de errores.
- Con *&1* indicamos que la redirección de la salida de errores se hará al fichero al que se haya redirigido la salida estándar (descriptor *1*).

4.4. Conceptos

4.4.1. i-nodos

Los i-nodos son una especie de indice que nos indica donde está localizado un determinado fichero dentro de un sistema de ficheros.

Aunque se tiende a asociar ficheros con i-nodos, un i-nodo no es un fichero.

importante
Todo fichero tiene asociado un i-nodo.

Un i-nodo contiene toda la información referente al fichero.

Un i-nodo también puede contener información sobre enlaces simbólicos, sockets y dispositivos especiales.

importante
Cuando creamos un sistema de ficheros lo creamos con una cantidad de i-nodos. Si se acaban los i-nodos no podremos escribir en el disco aunque nos quede espacio libre.

4.4.2. El *Virtual File System* o *VFS*

Aunque pueda parecer sencillo el hecho de acceder a diferentes tipos de sistemas de ficheros no lo es tanto ya que cada sistema de ficheros tiene sus peculiaridades.

GNU/Linux utiliza el *Virtual File System* para acceder a los diferentes sistemas de ficheros. El *VFS* es un sistema de ficheros genérico.

La forma más sencilla de explicarlo es recurrir a la programación orientada a objetos.

El *VFS* lo podemos considerar como una clase y los diferentes sistemas de ficheros como clases derivadas.

Todos los sistemas de ficheros comparten una serie de características comunes que son las heredadas del *VFS* y luego las suyas propias.

4.4.3. El *Buffer Cache*

El *Buffer Cache* es una memoria intermedia en la que se almacenan temporalmente las operaciones de escritura. Estas operaciones no se hacen de forma inmediata al sistema de ficheros. De este modo se mejora el rendimiento ya que no se penalizan otras acciones del sistema, como la interacción del usuario, por las operaciones de escritura.

Serán los algoritmos de multitarea los que decidan cuando van escribiendo esa información al disco.

Para el usuario todo esto es transparente y de cara a el es como si los datos se hubieran escrito al sistema de ficheros.

Más adelante veremos que es necesario montar un sistema de ficheros para hacerlo accesible. Debido a que los datos no se escriben al momento en el sistema de ficheros si quitamos un dispositivo físico del sistema sin que se hayan escrito los datos los perderemos. Se produce entonces una *inconsistencia en el sistema de ficheros*.

Para retirar un dispositivo es necesario desmontarlo, en ese momento se da prioridad a las operaciones de escritura pendientes y se escriben al sistema de ficheros. Una vez realizadas todas el sistema de ficheros es desmontado (ya no es accesible) sin perdida de datos.

sugerencia
Es posible utilizar **sync** para sincronizar un sistema de ficheros con el *Buffer Cache*.

4.4.4. Sistemas de ficheros *transaccionales* o de *journaling*

Estos sistemas de ficheros fueron desarrollados para ser tolerantes a fallos. El ejemplo típico es cuando se va la luz mientras estamos trabajando. Con sistemas de ficheros no *transaccionales* al iniciar de nuevo el ordenador el sistema de ficheros se tendrá que chequear. Con los sistemas de ficheros *transaccionales* el inicio es mucho más rápido ya que poseen la información necesaria para hacer la recuperación de forma rápida y precisa.

Todos los ficheros tienen asociados los siguientes datos:

- El contenido del fichero, los datos.
- Los datos referentes al fichero, tamaño, fecha, nombre, lugar dentro del sistema de ficheros en el que se encuentra, ...

En los sistemas de ficheros tradicionales al realizar una operación de escritura se modifican directamente los metadatos, pero no los datos que se irán modificando poco a poco. Si se va la luz, por ejemplo, cuando el sistema arranque los metadatos no coincidirán con los datos (*inconsistencia en el sistema de ficheros*), teniendo entonces una perdida de datos.

Los sistemas de ficheros *transaccionales* disponen de dos zonas para solucionar este problema:

1. Una zona de datos, donde se almacenarán tanto el contenido de los ficheros como los metadatos.
2. Una zona de *log* o *diario* donde es núcleo va registrando los cambios realizados y los pendientes.

Cuando el sistema actualiza los datos de un fichero, borra de la zona de *log* la información referente a los cambios realizados. De esta forma cuando se monta un sistema de ficheros *transaccional* si hay modificaciones pendientes en la zona de *log* se actualiza el sistema de ficheros sin perdida de datos.

Sistemas de ficheros *transaccionales*:

- *ext3*.
- *ReiserFS*.
- *JFS* de IBM.
- *XFS* de SGI.

4.4.5. Sistemas de ficheros de *acceso concurrente*

Los sistemas de ficheros de *acceso concurrente* son aquellos sistemas de ficheros que permiten el acceso simultaneo al sistema de ficheros a más de una máquina al mismo tiempo.

Estos sistemas de ficheros no se acceden por red ethernet como pudiera ser exportado por *NFS* o *CIFS*.

Estos sistemas de ficheros se suelen utilizar en redes *SAN* y son típicos de clusters en configuraciones activo/activo.

Cuando tenemos un cluster en configuración activo/activo con varios nodos todos los nodos estarán accediendo simultaneamente al sistema de ficheros. Es necesario coordinar mediante bloqueos los accesos de escritura para evitar corrupciones de datos.

Los sistemas de ficheros de *acceso concurrente* que soporta *GNU/Linux* son los siguientes:

- *GFS* o *G*lobal *F*ile *S*ystem es el sistema de ficheros de *acceso concurrente* que *Red Hat* compró a *Sixtina* para su inclusión en *Red Hat Cluster Suite*.
- *GPFS* o *G*eneral *P*aralell *F*ile *S*ystem es el sistema de ficheros de *acceso concurrente* de *IBM*.
- *OCFS* o *O*racle *C*luster *F*ile *S*ystem. Es el sistema de ficheros de *acceso concurrente* de Oracle que utiliza en *Oracle RAC*.

4.5. Sistemas de ficheros

Ya hemos dicho que en los sistemas UNIX like todo son ficheros.

Al contrario que en otros sistemas en los sistemas UNIX los ficheros están organizados de una forma rigurosa pero flexible:

- Mediante una organización jerarquica de directorios.
- Mediante sistemas de ficheros.

Un sistema de ficheros es un espacio en disco, bien sea una partición o un disco en su totalidad, en el cual se almacenan ficheros.

El estructurar la instalación de un sistema en sistemas de ficheros tiene las siguientes ventajas:

- Cuanto más pequeño sea un sistema de ficheros menos probable es una corrupción de datos.
- Si un sistema de ficheros se llena eso sólo perjudicará a las aplicaciones que escriban en ese sistema de ficheros. Si unicamente hay un sistema de ficheros todo el sistema se verá afectado.
- Permite una mayor estructuración y control.

- Ahorro en recursos.

  En los primeros tiempos de la informática los recursos eran muy caros. Para ahorrar costes se recurría a tener un único sistema de ficheros y compartirlo por *nfs* entre todas las máquinas que lo necesitaran.

  Esto pasaba, por ejemplo, con el directorio `/usr/bin`. Además también simplificaba la administración porque de esta forma es muy fácil garantizar que todas las máquinas tienen las mismas versiones de software instaladas.

  Pero esto también tenía sus problemas y es que si la máquina que exporta el sistema de ficheros deja de funcionar todas lo harán.

- Una mayor flexibilidad para la gestión de cuotas.

Esta estructuración también tiene sus inconvenientes:

- Se requiere de una buena planificación a la hora de elegir cuantos sistemas de ficheros vamos a montar y su tamaño.

importante
Cuando dividimos el sistema en varias particiones para asegurar que pueda arrancar en caso de problemas con los sistemas de ficheros es recomendable que los directorios `/bin/`, `/sbin/`, `/dev/`, `/etc/`, `/lib/` y `/root` esten en el *root filesystem*.

4.6. Particiones

Para poder utilizar un dispositivo físico, un disco duro, es necesario crear particiones en el.

En un disco duro se pueden tener los siguientes tipos de particiones:

- Primarias.
- Extendidas.
- Lógicas.

A la hora de crear particiones tenemos que tener en cuenta las siguientes limitaciones:

- El número total de particiones en un dispositivo físico será como mucho de 15.
- Sólo puede haber cuatro particiones primarias en un dispositivo físico.
- Sólo puede haber una partición extendida por dispositivo físico.
- La partición extendida cuenta como una primaria.
- Las particiones lógicas se incluyen siempre dentro de la extendida.

4.6.1. Particiones primarias

Estas particiones son las utilizamos cuando queremos que un sistema operativo arranque desde ella.

Como ya hemos dicho puede haber como mucho un máximo de cuatro por cada dispositivo físico (disco duro).

importante
Estas particiones siempre se numeran del uno al cuatro.

4.6.2. Particiones extendidas

Estas particiones son un contenedor en el cual se incluirán las particiones lógicas.

Estas particiones se consideran primarias y tienen la mismas limitaciones.

4.6.3. Particiones lógicas

Desde estas particiones no se puede arrancar un sistema operativo.

importante
Estas particiones se numeran del cinco al quince.

4.7. Tipos de dispositivos físicos

Existen diferentes tipos de discos físicos y a cada uno se le referencia de un modo diferente. El nombre del fichero correspondiente a ese dispositivo tiene un nombre diferente dependiendo del tipo de dispositivo que sea.

Existen diferentes tipos de dispositivos y para poder acceder a cada uno de ellos es necesario que el núcleo tenga soporte para dichos dispositivos.

4.7.1. Dispositivos IDE

Estos dispositivos reciben el nombre `/dev/hd?n`:

- *?* es una letra:
  - *a* es el canal primario del primer IDE.
  - *b* es el canal secundario del primer IDE.
  - *c* es el canal primario del segundo IDE.
  - *d* es el canal secundario del segundo IDE.
- *n* es el número de la partición dentro del dispositivo físico.

4.7.2. Dispositivos SCSI

La forma en la que se nombran los discos SCSI es la misma utilizada para:

- Memorias y discos USB.
- Discos SATA.
- Discos SAN (*S*torage *A*rea *N*etwork).

Estos dispositivos reciben el nombre `/dev/sd?n`:

- *?* son una o varias letras: *a*, *b*, ..., *z*, *aa*, *ab*, ..., *az*, *ba*, ... hasta un límite de 256 (como mucho).
  Estas letras se van asignando por orden alfabetico según se vayan descubriendo los dispositivos.
- *n* es el número de la partición dentro del dispositivo físico.

aviso
Con los dispositivos removibles y los discos SAN existe el problema de que el nombre del fichero que representa al disco físico no siempre va a ser el mismo.
Para un usuario esto no deja de se un engorro. Pero para un servidor esto es un problema bastante grande (que tiene solución).

4.7.3. Disqueteras

A las disqueteras nos referiremos como `/dev/fdn`:

- *n* es un número empezando en cero para la primera y siguiendo en orden ascendente.

4.7.4. Unidades de cinta

Podemos encontrar varios tipos de unidades de cinta:

- `/dev/stn` n-ésima unidad de cinta SCSI.
- `/dev/ftn` n-ésima unidad de cinta.

4.8. Acceso a sistemas de ficheros

Para tener acceso a los diferentes sistemas de ficheros es necesario:

- Soporte en el núcleo para el tipo de dispositivo físico.
- Soporte en el núcleo para el sistema de ficheros que hay en el dispositivo.
- Montar el sistema de ficheros.

4.8.1. El comando mount

El comando **mount** se utiliza para montar los sistemas de ficheros. Es decir para hacerlos accesibles desde el sistema.

En principio sólo es *superusuario* o *root* puede montar sistemas de ficheros en el sistema.

Para montar un sistema de ficheros:

```
[root@sal]# mount -t ext3 /dev/sda5 /media/removible
```

- *-t ext3* es el tipo de sistema de ficheros que reside en el dispositivo.
- */dev/sda5* dispositivo que se quiere montar.
- */media/removible* punto de montaje en el que se montará.

importante
Es posible especificar opciones de montaje mediante el uso de *-o* tales como sólo lectura, ...

Algunos tipos de sistemas de ficheros:

- *ext2*
- *ext3*
- *iso9660*
- *reiserfs*
- *xfs*
- *jfs*
- *nfs*
- *cifs*
- *vfat*
- *msdos*

Existen algunas opciones interesantes a la hora de montar un sistema de ficheros. Algunas de ellas son comunes a todos ellos, mientras que otras dependen del sistema de ficheros:

- *-o loop* permite el montar imágenes ISO a través del dispositivo de loopback:

```
[root@sal]# mount -t iso9660 debian-sarge-dvd1.iso /media/iso -o loop
```

- *-o rw* monta un sistema de ficheros en modo lectura/escritura.
- *-o ro* monta un sistema de ficheros en modo sólo lectura.
- *-o suid* permite la ejecución de SUIDs en el sistema de ficheros.
- *-o nosuid* no permite la ejecución de SUIDs en el sistema de ficheros.
- *-o exec* permite la ejecución de comandos en el sistema de ficheros.
- *-o noexec* no permite la ejecución de comando en el sistema de ficheros.
- *-o remount* permite montar un sistema de ficheros ya montado con otras opciones diferentes.
- *-o owner* permite a un usuario sin privilegios montar un sistema de ficheros si es el propietario del dispositivo físico.
- *-o nouser* no permite a los usuarios sin privilegios el montar sistemas de ficheros.
- *-o user* permite a cualquier usuario montar y desmontar el sistema de ficheros.

4.8.2. El comando umount

El comando **umount** se utiliza para desmontar sistemas de ficheros:

```
[root@sal]# umount /media/iso
```

Desmontaría el sistema de ficheros montado en `/media/iso`.

Sólo el *superusuario* o *root* puede desmontar sistemas de ficheros.

importante
Es posible permitir a usuarios sin privilegios montar sistemas de ficheros. En este caso el usuario que montó el sistema de ficheros podrá desmontarlo también.

aviso

Para poder desmontar un sistema de archivos es necesario que ningún recurso del sistema este usando dicho dispositivo:

```
[root@sal]# pwd
/media/iso
[root@sal]# umount /media/iso
umount: /media/iso: dispositivo ocupado
[root@sal]#
```

aviso

Bajo ningún concepto se debe desconectar ningún dispositivo de almacenamiento del sistema sin desmontarlo previamente.
Cuando se hace una operación de escritura sobre un dispositivo, rara vez se hace directamente al dispositivo. Normalmente se escribe en una zona intermedia denominada *buffer cache* y son los algoritmos de multitarea los que deciden cuando se va a ir escribiendo esa información al dispositivo.
Cuando se hace un **umount** de un dispositivo se priorizan las escrituras pendientes sobre ese dispositivo (se hace un **sync** sobre el dispositivo). Por este motivo si se desconecta un dispositivo del sistema sin haberlo desmontado previamente se perderan datos y se dice entonces que hay una inconsistencia en el sistema de archivos.
Con sistemas de archivos *transaccionales* o de *journaling* como *ext3* o *Reiser* es posible, en principio, recuperar esa información.

4.8.3. El fichero de configuración `/etc/fstab`

Este fichero tiene dos propositos:

1. Especificar que sistemas de ficheros se montan en el arranque.
2. Indicar las opciones de montaje de los sistemas de ficheros.

sugerencia

Es posible montar un sistema de archivos especificando unicamente el punto de montaje si está en el fichero `/etc/fstab`. Se montará con las opciones especificadas en dicho fichero.

Un `/etc/fstab` típico:

```
LABEL=/     /             ext3          defaults       1 1
none        /proc         proc          defaults       0 0
/dev/hda5   swap          swap          defaults       0 0
/dev/hdc    /media/cdrom  iso9660       noauto,user,ro 0 0
/dev/fd0    /media/floppy auto          noauto,user    0 0
/dev/hda6   /media/vfat   vfat          noauto,user    0 0
/dev/sda1   /media/scsi   reiserfs      auto           0 0
```

Este fichero consta de seis columnas:

1. La primera columna es el dispositivo físico o la etiqueta del sistema de ficheros.
2. La segunda columna es el punto de montaje.
3. La tercera columna es el sistema de ficheros que hay en el dispositivo.
4. La cuarta columna son las opciones de montaje.

5. La quinta columna es información para **dump**. Si es un cero no es necesario hacer un dump del sistema de ficheros.
   En caso de no encontrar nada se asume que es un cero.
6. La sexta columna indica a **fsck** el orden en el que chequeará los sistemas de ficheros al arrancar.
   Si hay un cero no chequeará el sistema de ficheros.

---

sugerencia
Si se utiliza *user* como una de las opciones entonces se permite montar el sistema a cualquier usuario con las opciones indicadas en `/etc/fstab` pasandole a **mount** el punto de montaje especificado en `/etc/fstab` unicamente.

4.8.4. El fichero `/proc/partitions`

Este fichero nos muestra las particiones que tenemos en el sistema:

```
[root@sal]# cat /proc/partitions

major minor  #blocks  name

   3     0  195360984 hda
   3     1  192418506 hda1
   3     2          1 hda2
   3     5    2939863 hda5

[root@sal]#
```

4.8.5. El fichero `/proc/filesystems`

Este fichero nos muestra los filesystems con los que puede trabajar el núcleo en ese momento.

```
[root@sal]# cat /proc/filesystems

nodev   sysfs
nodev   rootfs
nodev   bdev
nodev   proc
nodev   securityfs
nodev   sockfs
nodev   pipefs
nodev   futexfs
nodev   tmpfs
nodev   inotifyfs
nodev   eventpollfs
nodev   devpts
        cramfs
nodev   ramfs
nodev   mqueue
nodev   usbfs
        ext3

[root@sal]#
```

4.8.6. El fichero `/etc/mtab`

Este fichero nos muestra los filesystems que tenemos montados en el sistema y las opciones con las que fueron montados.

```
[root@sal]# cat /etc/mtab
/dev/hda1 / ext3 rw,errors=remount-ro 0 0
proc /proc proc rw,noexec,nosuid,nodev 0 0
/sys /sys sysfs rw,noexec,nosuid,nodev 0 0
udev /dev tmpfs rw,mode=0755 0 0
devshm /dev/shm tmpfs rw 0 0
devpts /dev/pts devpts rw,noexec,nosuid,gid=5,mode=620 0 0
usbfs /proc/bus/usb usbfs rw,noexec,nosuid,nodev 0 0
[root@sal]#
```

4.9. Creación de sistemas de ficheros

Para crear un sistema de ficheros lo primero que tenemos que hacer es crear una partición. Para ello podemos utilizar el comando **fdisk** o **parted**.

4.9.1. Los sistemas de ficheros *ext2/ext3*

4.9.1.1. El comando mkfs

Este comando es utilizado para crear los sistemas de ficheros *ext2* y *ext3*, entre otros. Para más información mirar la página del manual.

Su sintaxis es muy sencilla:

```
[root@sal]# mkfs -t ext2 /dev/fd0
mke2fs 1.23, 15-Aug-2001 for EXT2 FS 0.5b, 95/08/09
Filesystem label=
OS type: Linux
Block size=1024 (log=0)
Fragment size=1024 (log=0)
184 ubidesm 1440 blocks
72 blocks (5.00\%) reserved for the super user
First data block=1
1 block group
8192 blocks per group, 8192 fragments per group
184 inodes per group

Writing inode tables: done
Writing superblocks and filesystem accounting information: done

This filesystem will be automatically checked every 20 mounts or
180 days, whichever comes first. Use tune2fs -c or -i to override.
[root@sal]#
```

La opción más interesante es *-c* para que busque bloques en mal estado durante el formateo.

---

sugerencia
Para los tipos de sistemas de ficheros manejados por **mkfs** existe el comando **mkfs.tipo** donde *tipo* es *ext2*, ... y su uso es igual al de **mkfs** sólo que no es necesario especificar el tipo de sistema de ficheros a crear.

4.9.1.2. Conversión de sistemas de ficheros en *ext2* a *ext3*

En GNU/Linux los sistemas de ficheros más extendidos son *ext2* y *ext3*.

En realidad *ext3* se puede considerar como *ext2* con algunas mejoras. La más significativa de ellas es el *journaling*.

Para convertir un sistema de ficheros *ext2* a *ext3*:

```
[root@sal]# tune2fs -j /dev/sda1
```

4.9.1.3. El superbloque y tune2fs

El superbloque es bloque especial que contiene información sobre el sistema de ficheros. Para los sistemas de ficheros *ext2/ext3* se puede ver o modificar con **tune2fs**:

```
[root@sal]# tune2fs -l /dev/hda2
tune2fs 1.40-WIP (02-Oct-2006)
Filesystem volume name:   <none>
Last mounted on:          <not available>
Filesystem UUID:          85314ef9-3429-42d2-b8bc-41b51b328ffa
Filesystem magic number:  0xEF53
Filesystem revision #:    1 (dynamic)
Filesystem features:      has_journal resize_inode dir_index filetype needs_recovery  ↩
    sparse_super large_file
Default mount options:    (none)
Filesystem state:         clean
Errors behavior:          Continue
Filesystem OS type:       Linux
Inode count:              24068096
Block count:              48104626
Reserved block count:     2405231
Free blocks:              43311411
Free inodes:              23958460
First block:              0
Block size:               4096
Fragment size:            4096
Reserved GDT blocks:      1012
Blocks per group:         32768
Fragments per group:      32768
Inodes per group:         16384
Inode blocks per group:   512
Filesystem created:       Sun Mar 25 06:10:23 2007
Last mount time:          Thu Apr  5 04:55:15 2007
Last write time:          Thu Apr  5 04:55:15 2007
Mount count:              9
Maximum mount count:      26
Last checked:             Sun Mar 25 06:10:23 2007
Check interval:           15552000 (6 months)
Next check after:         Fri Sep 21 06:10:23 2007
Reserved blocks uid:      0 (user root)
Reserved blocks gid:      0 (group root)
First inode:              11
Inode size:               128
Journal inode:            8
Default directory hash:   tea
Directory Hash Seed:      1ff4a0e4-fceb-45c1-93f8-5e9799e39496
Journal backup:           inode blocks
[root@sal]#
```

Las opciones más interesantes que podemos realizar:

- *-c n* indicar después de cuantas operaciones de montaje se ha de realiza el chequeo del sistema de ficheros.
- *-j* añadir la zona de *journaling* a un sistema de ficheros *ext2* para convertirlo en *ext3*.
- *-C n* cambiar el número de veces que el sistema ha sido montado. Si se estable a un valor mayor del indicado por *-c n* en el siguiente montaje se chequeará de forma automática.

- *-g gid* indicar el GID del grupo para el que se reservará espacio en el sistema de ficheros.
- *-i n [d|w|m]* establece el periodo máximo de tiempo tras el cual se chequeará el sistema de ficheros. Independientemente de si se ha llegado al número máximo de operaciones de montaje.
- *-L label* establece la etiqueta del sistema de ficheros.
- *-m n* establece el tanto por ciento de bloques que se reservan.
- *-r n* establece el número de bloques que se reservan.
- *-u uid* establece el usuario para el que se reservará espacio.

4.9.1.4. El comando mke2fs

La función de este comando es la de crear sistemas de ficheros *ext2/ext3* y su uso es preferible al de **mkfs**.

Se utiliza el fichero `/etc/mke2fs.conf` para determinar los valores por defecto cuando se crea un sistema de ficheros.

aviso
No modificar `/etc/mke2fs.conf` a menos que se sepa lo que se está haciendo.

4.9.1.5. Recuperación de sistemas de ficheros *ext2/ext3*

Cuando nos encontremos errores en los sistemas de ficheros tendremos que chequearlos y para ello tendrán que estar desmontados.

El comando utilizado para chequear estos sistemas de ficheros es **e2fsck**.

Algunas opciones interesantes:

- *-b n* donde *n* indica una copia del superbloque.

  En los sistemas de ficheros *ext2/ext3* se hacen copias de seguridad del superbloque repartidas por todo el sistema de ficheros (ver página del manual).
- *-c* se utiliza el comando **badblocks** para localizar bloques defectuosos y marcarlos como tales.
- *-f* forzar el chequeo aunque el sistema de ficheros parezca limpio.
- *-p* reparará de forma automática el sistema de ficheros si se puede hacer de forma segura.

sugerencia
Al igual que en el caso de **mke2fs** existía **mkfs** para chequear sistemas de ficheros también se puede utilizar el comando **fsck**.

4.9.1.6. El comando badblocks

Este comando se utiliza para localizar bloques defectuosos en un sistema de ficheros.

La forma más sencilla de utilizarlo es:

```
[root@sal]# badblocks /dev/sda1
```

sugerencia
Podemos guardar la lista de bloques defectuosos en un fichero utilizando *-o filename*.

4.9.2. El sistema de ficheros *ReiserFS*

ReiserFS es otro sistema de ficheros *transaccional*.

sugerencia
Será necesario instalar las utilidades de administración de dicho sistema de ficheros ya que no se suelen instalar por defecto.

Las principales utilidades de administración son:

- *mkreiserfs* para crear sistemas de ficheros.
- *reiserfsck* para chequear sistemas de ficheros.
- *reiserfstune* para ajustar los parámetros del sistema de ficheros.
- *resize_reiserfs* para redimensionar sistemas de ficheros.

4.9.3. El sistema de ficheros *JFS*

Este sistema de ficheros lo desarrolló *IBM* y al igual que *ext3* y *ReiserFS* es un sistema de ficheros *transaccional*.

sugerencia
Será necesario instalar las utilidades de administración de dicho sistema de ficheros ya que no se suelen instalar por defecto.

Las principales utilidades de administración son:

- *jfs_mkfs* para crear sistemas de ficheros.
- *jfs_fsck* para chequear sistemas de ficheros.
- *jfs_tune* para ajustar los parámetros del sistema de ficheros.

4.9.4. El sistema de ficheros *XFS*

Este sistema de ficheros lo desarrolló *SGI* y al igual que *ext3* y *ReiserFS* es un sistema de ficheros *transaccional*.

sugerencia
Será necesario instalar las utilidades de administración de dicho sistema de ficheros ya que no se suelen instalar por defecto.

Las principales utilidades de administración son:

- *mkfs.xfs* para crear sistemas de ficheros.
- *xfs_check* para chequear sistemas de ficheros.
- *xfs_repair* para reparar sistemas de ficheros.
- *xfs_admin* para ajustar los parámetros del sistema de ficheros.
- *xfs_freeze* congela el acceso al sistema de ficheros. Util para la creación de snapshots.
- *xfs_growfs* redimensionar el sistema de ficheros.
- *xfs_quota* manejo de cuotas.

4.10. Obtención de información sobre los sistemas de ficheros

4.10.1. El comando du

Este comando trabaja a nivel de directorios. En caso de no pasarle ningún argumento nos muestra el tamaño que utiliza en disco el directorio actual y sus subdirectorios. Por defecto muestra la información en bloques y si utilizamos el flag *-h* nos mostrará la información en un formato más comprensible.

```
[pcm@sal]$ du
245     ./Adm/html
486     ./Adm
16539   ./Doc/cluster
18256   ./Doc
4       ./Software-Cientifico
16      ./Clustering/dia
125     ./Clustering/html
44      ./Clustering/images
269     ./Clustering
19014   .
[pcm@sal]$
```

sugerencia
Si le pasamos como argumento un directorio nos dará la información sobre dicho directorio.

sugerencia
Si unicamente queremos conocer el espacio en disco deberemos utilizar el flag *-hs*.

4.10.2. El comando df

Ese comando nos da información sobre los sistemas de ficheros montados en el sistema:

```
[pcm@sal]$ df
S.ficheros         Bloques de 1K   Usado    Dispon Uso% Montado en
/dev/hdg1               459143     82646    352000  20% /
tmpfs                  1038484         0   1038484   0% /lib/init/rw
udev                     10240        44     10196   1% /dev
tmpfs                  1038484         0   1038484   0% /dev/shm
/dev/mapper/system_vg-home_lv
                       2097084   1633876    463208  78% /home
/dev/mapper/system_vg-opt_lv
                        511980     32840    479140   7% /opt
/dev/mapper/system_vg-tmp_lv
                       1023964     32852    991112   4% /tmp
/dev/mapper/system_vg-usr_lv
                       5242716   2528076   2714640  49% /usr
/dev/mapper/system_vg-var_lv
                       2097084    575252   1521832  28% /var
/dev/mapper/system_vg-software_lv
                       1048540     32840   1015700   4% /mnt/software
/dev/mapper/system_vg-ftp_lv
                      52427196  18240412  34186784  35% /media/ftp
[pcm@sal]$
```

sugerencia
Podemos ver la información relativa a un único sistema de ficheros si se lo pasamos como argumento.

Por defecto muestra información en bloques. Podemos hacer que salga la información en un formato más comprensible utilizando el flag *-h*:

```
[pcm@sal]$ df -h
S.ficheros          Tamaño Usado  Disp Uso% Montado en
/dev/hdg1             449M   81M  344M  20% /
tmpfs                1015M     0 1015M   0% /lib/init/rw
udev                   10M   44K   10M   1% /dev
tmpfs                1015M     0 1015M   0% /dev/shm
/dev/mapper/system_vg-home_lv
                      2,0G  1,6G  451M  78% /home
/dev/mapper/system_vg-opt_lv
                      500M   33M  468M   7% /opt
/dev/mapper/system_vg-tmp_lv
                     1000M   33M  968M   4% /tmp
/dev/mapper/system_vg-usr_lv
                      5,0G  2,5G  2,6G  49% /usr
/dev/mapper/system_vg-var_lv
                      2,0G  562M  1,5G  28% /var
/dev/mapper/system_vg-software_lv
                      1,0G   33M  992M   4% /mnt/software
/dev/mapper/system_vg-ftp_lv
                       50G   18G   33G  35% /media/ftp
[pcm@sal]$
```

Podemos hacer que salga la información sobre los inodos utilizando el flag *-i*:

```
[pcm@sal]$ df -i
S.ficheros           Nodos-i NUsados NLibres NUso% Montado en
/dev/hdg1             245280   11612  233668    5% /
tmpfs                 224347       2  224345    1% /lib/init/rw
udev                  224347    1430  222917    1% /dev
tmpfs                 224347       1  224346    1% /dev/shm
/dev/mapper/system_vg-home_lv
                           0       0       0    -  /home
/dev/mapper/system_vg-opt_lv
                           0       0       0    -  /opt
/dev/mapper/system_vg-tmp_lv
                           0       0       0    -  /tmp
/dev/mapper/system_vg-usr_lv
                           0       0       0    -  /usr
/dev/mapper/system_vg-var_lv
                           0       0       0    -  /var
/dev/mapper/system_vg-software_lv
                           0       0       0    -  /mnt/software
/dev/mapper/system_vg-ftp_lv
                           0       0       0    -  /media/ftp
[pcm@sal]$
```

4.11. Cuotas en *ext2/ext3*

Es posible establecer cuotas de espacio en disco tanto por usuario como por grupo para evitar que un determinado usuario o grupo monopolice el espacio en disco.

Esta característica tiene que estar soportada dentro del núcleo.

El espacio en disco es un recurso finito y una mala gestión de su uso puede provocar una denegación de servicio.

Será necesario controlar la actividad de los usuarios para evitar un mal uso del espacio en disco.

Aunque el espacio en disco pueda parecer asequible a los usuarios en realidad es bastante caro. Ya que el tener espacio desaprovechado implica:

- Un mayor coste en tiempo y recursos de almacenamiento debido a las políticas de backup.
- Un mayor coste en tiempo a la hora de restaurar backups.
- Hoy en día el espacio en disco se suele asignar en *SAN* debido a su verstilidad. El uso de tecnologías *SAN* es de un alto coste.

4.11.1. ¿En qué sistemas de ficheros podemos establecer cuotas de usuario?

Podemos establecer cuotas de espacio en disco en todos los sistemas de ficheros que aparezcan en `/etc/fstab`.

Debido a la posibilidad de realizar una instalación en varios sistemas de ficheros podemos optimizar el uso de cuotas para los usuarios y grupos en cada sistema de fichero.

4.11.2. Cuotas *hard*

Las cuotas *hard* establecen la cantidad máxima de espacio que se puede utilizar y no se pueden sobrepasar a menos que el administrador las cambie. Estas cuotas se pueden establecer:

- Por usuario. Una vez superado el usuario no podrá escribir en el sistema de ficheros.

  sugerencia
  Al contrario que en otros sistemas un usuario que haya sobrepasado la cuota sí podrá borrar ficheros.

- Por grupo. Una vez superado ningún usuario del grupo podrá escribir en el sistema de ficheros, a pesar de que no haya alcanzado su cuota como usuario.

4.11.3. Cuotas *soft*

Las cuotas *soft* establen el umbral para avisar a los usuarios o grupos de que están llegando al límite máximo o *cuota hard*. Cuando se llega a esta cuota cada vez que un usuario que la ha sobrepasado escribe en el sistema de ficheros le aparece un mensaje en la terminal recordandoselo. Estas cuotas se pueden establecer:

- Por usuario.
- Por grupo.

4.11.4. El periodo de gracia

Cuando se sobrepasa la *cuota soft* se entra en el *periodo de gracia*. Una vez terminado el periodo no le es permitido al usuario o grupo escribir en el sistema de ficheros hasta que libere el espacio necesario para estar por debajo de la *cuota soft*.

Este periodo se puede especificar en meses, semanas, dís, horas, minutos o segundos.

4.11.5. Pasos previos a la activación de las cuotas

Será necesario que el núcleo este activado con soporte para cuotas.

Como se podría esperar sólo es *superusuario* o *root* podrá establecer las cuotas.

Para cada sistema de ficheros en el que queramos establecer las cuotas deberemos hacer lo siguiente:

1. Añadir en el fichero `/etc/fstab` las opciones *usrquota* para habilitar las cuotas para usuarios y *grpquota* para los grupos.
2. Crear los ficheros `quota.user` y `quota.group` en la raíz del sistema de ficheros y establecer los permisos adecuados. Supongamos que queremos establecer las cuotas de usuario y grupo para el sistema de ficheros */dev/sda5* que está montado en `/home`:

```
[root@sal]# touch /home/quota.user
[root@sal]# touch /home/quota.group
[root@sal]# chmod 400 /home/quota.*
```

importante
Estos ficheros contendran datos binarios y no texto.

3. Tendremos que inicializar las bases de datos que van a almacenar la información relativa a las cuotas en los ficheros que hemos creado anteriormente:

```
[root@sal]# quotacheck -avug
quotacheck: Scanning /dev/sda5 [/home] done
quotacheck: Checked 79 directories and 657 files
[root@sal]#
```

   Las opciones que le hemos pasado a **quotacheck**:

   - *a* realiza la comprobación para todos los sistemas de ficheros con cuotas.
   - *v* modo verbose.
   - *u* realiza la comprobación para las cuotas de usuario.
   - *g* realiza la comprobación para las cuotas de grupo.

4. Activamos el sistema de quotas:

```
[root@sal]# quotaon -a
[root@sal]#
```

4.11.6. Estableciendo cuotas

Una vez configurado y arrancado el sistema de cuotas tendremos que establecer las cuotas para los diferentes usuarios y grupos.

Las cuotas se establecen por bloques e inodos. A menos que se haya jugueteado con las opciones la crear un sistema de ficheros cada bloque equivaldráa, normalmente, a 1 KB (1.24 bytes).

Utilizaremos el comando **edquota** para establecer las cuotas. Este comando accede a los ficheros `quota.user` y `quota.group` creando un fichero temporal en `/tmp` editandolo por defecto con *vi* a menos que se especifique otro editor en las variables de entorno *EDITOR* o *VISUAL*. Algunos de los flags que podemos utilizar son:

- *u* que se utiliza para editar las cuotas de disco de los usuarios. Si se especifica la opción *g* esta opción es ignorada.

- *g* que se utiliza para editar las cuotas de disco de los grupos.

Para cambiar las cuotas del usuario *pcm* deberemos:

```
[root@sal]# edquotacheck -u pcm
```

A continuación se accederá a los ficheros de cuotas y se creará en `/tmp` un fichero con datos que será editado y aparecerá algo como esto:

```
Disk quotas for user pcm (uid 1000):
Filesystem     blocks   soft   hard   inodes   soft   hard
/dev/sda5        1084  15000  25000      732   2000   3500
```

Modificamos los valores para *hard* y *soft* y al grabar y salir se actualiza el sistema de cuotas con los nuevos datos.

importante
Los valores que aparecen en *blocks* y en *inodes* son los bloques e inodos que está utilizando el usuario en ese sistema de ficheros en ese momento.

Es posible establecer cuotas de usuario en línea de comando utilizando el comando **setquota**.

sugerencia
man setquota

4.11.7. Estableciendo el periodo de gracia

El periodo de gracia lo estableceremos con **edquota -t**. El procedimiento es el mismo que para establecer las cuotas de usuarios y grupos con **edquota**. Al ejecutar **edquota -t**:

```
[root@sal]# edquota -t
Grace period before enforcing soft limits for users:
Time units may be: days, hours, minutes, or seconds
Filesystem       Block grace period   Inode grace period
/dev/sda5        7days                7 days
```

Podemos establecer el periodo de gracia pro bloques o por inodos. Lo establecemos según nuestras necesidades y al salir grabando se actualizan las BBDD del sistema con los nuevos datos.

4.11.8. Iniciando y parando el sistema de cuotas

Una vez que hemos configurado los sistemas de ficheros sobre los cuales tendremos cuotas y hemos establecido dichas cuotas tendremos que arrancar el sistema de cuotas:

```
[root@sal]# quotaon -av
/dev/sda5 [/home]: group quotas turned on
/dev/sda5 [/home]: user quotas turned on
[root@sal]#
```

Los flags más habituales son:

- *a* que inicializa las cuotas en todos los sistemas de ficheros que las tienen activadas.
- *v* modo verbose.

- *u* unicamente inicializa las cuotas de usuario.
- *g* unicamente inicializa las cuotas de grupo.

Es posible inicializar las cuotas sobre un determinado sistema de ficheros:

```
[root@sal]# quotaon -v /opt
/dev/sda6 [/opt]: group quotas turned on
/dev/sda6 [/opt]: user quotas turned on
[root@sal]#
```

Podemos parar el sistema de cuotas con el comando **quotaoff**.

sugerencia
man quotaoff

4.11.9. Chequeando el sistema de cuotas

Utilizaremos para ello el comando **quotacheck**. Este comando se utiliza para chequear y actualizar el uso de espacio en disco en los sistemas de ficheros y para repar los ficheros de cuotas `quota.user` y `quota.group`.

Por defecto sólo se comprueban las cuotas de usuario, si se quieren comprobar las de grupo habrá que utilizar *g*.

aviso
Cuando se chequeen las cuotas de disco es recomendable hacerlo con el sistema de cuotas parado.

aviso
Es aconsejable que cuando se arranque el sistema se comprueben los sistemas de ficheros con cuotas antes de inicializar el sistema de cuotas.

Los flags más habituales son:

- *a* comprueba todos los sistemas de ficheros montados con cuotas establecidas.
- *g* comprueba las cuotas de grupo. No se comprueban a menos que se use este flag.
- *i* trabaja en modo interactivo.
- *u* comprueba las cuotas de usuario. Acción por defecto.
- *v* modo verbose. Es aconsejable utilizar este flag.

4.11.10. Reporting de cuotas

Mediante el uso de **repquota** es posible obtener informes sobre el estado de las cuotas del sistema. Los flags más significativos son:

- *a* informe sobre todos los sistemas de ficheros con cuotas presentes en `/etc/fstab`.
- *g* informe sobre las cuotas de grupo.

- *u* informe sobre las cuotas de usuario.

```
[root@sal]# repquota -a
*** Report for user quotas on device /dev/sda5
Block grace time: 7days; Inode grace time: 7days
                  Block limits                File limits
User       used   soft   hard   grace    used    soft   hard   grace
----------------------------------------------------------------------
root   --    20      0      0               4       0      0
pcm -- 10848  15000  25000             732    2000   3500
[root@sal]#
```

sugerencia
Es posible sacar informes sobre un único sistema de ficheros pasandole el punto de montaje o el dispositivo físico como argumento a **repquota**.

4.12. Atributos en sistemas de ficheros *ext2/ext3*

El usuario *root* puede acceder sin restricciones a todo el sistema y esto puede suponer un peligro ya que sin querer se puede borrar un fichero. Mediante el sistema de atributos se puede solucionar.

Ademas de permisos los ficheros también tienen atributos.

aviso
Esta característica tiene que estar compilada en el núcleo.

4.12.1. El comando chattr

Este comando se utiliza para cambiar los atributos de un fichero.

Algunos de los atributos que podemos establecer son:

- *no modificable* el fichero no se puede modificar, renombrar, ni hacer enlaces a un fichero con este atributo activado. Para activarlo utilizaremos el flag *-i*.

importante
Sólo el *root* puede establecer este atributo.

- *añadir* unicamente se puede añadir información al fichero. Para activarlo utilizaremos el flag *-a*.
- *borrado seguro* antés de borrar el fichero lo sobreescribe con ceros y lo guarda en disco. Para activarlo utilizaremos el flag *s*.

sugerencia
Utilizaremos "=" para asignar atributos, "+" para añadir atributos y "-" para quitarlos.

4.12.2. El comando lsattr

Este comando se utiliza para ver los atributos que tiene establecidos un fichero.

```
[pcm@sal]$ lsattr admlinux.xml
----i-------------- admlinux.xml
[pcm@sal]$
```

Vemos que el fichero `admlinux.xml` tiene activado el atributo de *no modificable* con lo cual no podrá ser modificado a menos que el *root* le quite ese atributo.

Capítulo 5

Gestión de sistemas de ficheros mediante *LVM*

LVM permite una mejor y más flexible administración de los sistemas de ficheros.

Mediante el uso de almacenamiento externo, *LVM* y sistemas de ficheros como *ext3*, *ReiserFS* y *XFS* que permiten redimensionar *en caliente* tenemos una posibilidades de crecimiento y gestión de recursos de almacenamiento practicamente ilimitadas.

importante
En este capítulo se pretende introducir *LVM* para familiarizar al alumno con los conceptos básicos del manejo de volumenes de discos. No se entrará a valorar las posibilidades de snapshots o clustering de *LVM*.

5.1. Volumenes físicos (physical volumes)

Los *volumenes físicos* son los dispositivos físicos de almacenamiento. En base a estos se establece todo el sistema de gestión de .

Para poder utilizar un disco físico o una partición con *LVM* es necesario iniciarlizarla:

```
[root@sal]# pvcreate /dev/sda
  Physical volume "/dev/sda" successfully created
[root@sal]#
```

5.1.1. Información y detección de volumenes físicos

Podemos utilizar el comando **pvscan** para buscar volumenes físicos:

```
[root@sal]# pvscan
  PV /dev/sda7   VG system_vg   lvm2 [107,59 GB / 26,13 GB free]
  Total: 1 [107,59 GB] / in use: 1 [107,59 GB] / in no VG: 0 [0   ]
[root@sal]#
```

También disponemos del comando **pvdisplay** que nos ofrece más información sobre los volumenes encontrados:

```
[root@sal]# pvdisplay
  --- Physical volume ---
  PV Name               /dev/sda7
  VG Name               system_vg
  PV Size               107,59 GB / not usable 0
  Allocatable           yes
  PE Size (KByte)       4096
  Total PE              27544
```

```
  Free PE               6689
  Allocated PE          20855
  PV UUID               EcJiMO-20Ve-QVAz-yZvE-jN5e-tGrH-RtwyHB
[root@sal]#
```

importante
El *UUID* es un identificador utilizado para señalar de forma única a cada volumen físico.

El comando *pvs* también nos ofrece información:

```
[root@sal]# pvs
  PV         VG        Fmt  Attr PSize   PFree
  /dev/sda7  system_vg lvm2 a-   107,59G 26,13G
[root@sal]#
```

LVM proporciona el comando **lvmdiskscan** que nos inidicará todos los discos visibles del sistema:

```
[root@sal]# lvmdiskscan
  /dev/ramdisk [       16,00 MB]
  /dev/dm-0    [       20,00 GB]
  /dev/ram     [       16,00 MB]
  /dev/sda1    [      101,94 MB]
  /dev/dm-1    [       20,00 GB]
  /dev/ram2    [       16,00 MB]
  /dev/sda2    [        4,00 GB]
  /dev/ram3    [       16,00 MB]
  /dev/root    [        1,50 GB]
  /dev/ram4    [       16,00 MB]
  /dev/ram5    [       16,00 MB]
  /dev/sda5    [        6,00 GB]
  /dev/ram6    [       16,00 MB]
  /dev/sda6    [        4,00 GB]
  /dev/ram7    [       16,00 MB]
  /dev/sda7    [        2,00 GB]
  /dev/ram8    [       16,00 MB]
  /dev/sda8    [        2,00 GB]
  /dev/ram9    [       16,00 MB]
  /dev/sda9    [        2,00 GB]
  /dev/ram10   [       16,00 MB]
  /dev/sda10   [       14,31 GB]
  /dev/ram11   [       16,00 MB]
  /dev/ram12   [       16,00 MB]
  /dev/ram13   [       16,00 MB]
  /dev/ram14   [       16,00 MB]
  /dev/ram15   [       16,00 MB]
  /dev/sdb     [       50,00 GB] LVM physical volume
  2 disks
  24 partitions
  1 LVM physical volume whole disks
  0 LVM physical volumes
[root@sal]#
```

5.1.2. Eliminación de volumenes físicos

Podemos eliminar volumenes físicos con el comando **pvremove**.

```
[root@sal]# pvremove /dev/sda
  Labels on physical volume "/dev/sda" successfully wiped
[root@sal]#
```

5.2. Grupos de volumen (volume groups)

Los grupos de volumen son el equivalente a discos duros virtuales. Es decir un grupo de volumen estará formado por uno o varios dispositivos físicos o particiones.

Un grupo de volumen se crea de la siguiente forma:

```
[root@sal]# vgcreate data_vg /dev/sda /dev/sdb /dev/sdc5
  Volume group "data_vg" successfully created
[root@sal]#
```

De esta forma habremos creado un grupo de volumen, disco virtual, que estará formado por los dispositivos físicos */dev/sda*, */dev/sdb* y la partición */dev/sdc5*.

Una vez hecho esto tendremos un directorio `/dev/data_vg/` en el cual se irán creando los ficheros de dispositivo que hacen referencia a las particiones lógicas que creemos dentro de este grupo de volumen.

5.2.1. Información y detección de grupos de volumen

Podemos utilizar el comando **vgscan** para encontrar grupos de volumen:

```
[root@sal]# vgscan
  Reading all physical volumes. This may take a while...
  Found volume group "data_vg" using metadata type lvm2
[root@sal]#
```

También disponemos del comando **vgdisplay** que nos ofrece más información sobre los grupos de volumen encontrados:

```
[root@sal]# vgdisplay
  --- Volume group ---
  VG Name               data_vg
  System ID
  Format                lvm2
  Metadata Areas        1
  Metadata Sequence No  6
  VG Access             read/write
  VG Status             resizable
  MAX LV                0
  Cur LV                2
  Open LV               1
  Max PV                0
  Cur PV                1
  Act PV                1
  VG Size               50,00 GB
  PE Size               4,00 MB
  Total PE              12799
  Alloc PE / Size       10240 / 40,00 GB
  Free  PE / Size       2559 / 10,00 GB
  VG UUID               JJ3bIX-tqSM-r8Qd-nt0b-jWm9-7O4n-YiZYBK
[root@sal]#
```

sugerencia
Podemos ver información detallada sobre el grupo de volumen utilizando el flag *-v*.

```
[root@sal]# vgdisplay -v
  --- Volume group ---
  VG Name               data_vg
  System ID
  Format                lvm2
  Metadata Areas        1
  Metadata Sequence No  6
  VG Access             read/write
  VG Status             resizable
  MAX LV                0
  Cur LV                2
  Open LV               1
  Max PV                0
  Cur PV                1
  Act PV                1
  VG Size               50,00 GB
  PE Size               4,00 MB
  Total PE              12799
  Alloc PE / Size       10240 / 40,00 GB
  Free  PE / Size       2559 / 10,00 GB
  VG UUID               JJ3bIX-tqSM-r8Qd-nt0b-jWm9-7O4n-YiZYBK

  --- Logical volume ---
  LV Name                /dev/data_vg/apache_lv
  VG Name                data_vg
  LV UUID                KCDc3t-jAHj-dNDK-I9qe-rF3B-oyya-Tg00JL
  LV Write Access        read/write
  LV Status              available
  # open                 1
  LV Size                20,00 GB
  Current LE             5120
  Segments               1
  Allocation             inherit
  Read ahead sectors     0
  Block device           253:0

  --- Logical volume ---
  LV Name                /dev/data_vg/mysql_lv
  VG Name                data_vg
  LV UUID                0m24gV-H9BC-MdbV-vSNZ-r57x-gx37-zUy22q
  LV Write Access        read/write
  LV Status              available
  # open                 0
  LV Size                20,00 GB
  Current LE             5120
  Segments               1
  Allocation             inherit
  Read ahead sectors     0
  Block device           253:1

  --- Physical volumes ---
  PV Name               /dev/sdd
  PV UUID               fKgSus-ts2y-vbR2-mTfy-YKPY-Aju1-C8SgV0
  PV Status             allocatable
  Total PE / Free PE    12799 / 2559
[root@sal]#
```

El comando **vgs** también nos ofrece información:

```
[root@sal]# vgs
  VG      #PV #LV #SN Attr   VSize  VFree
  data_vg   1   2   0 wz--n- 50,00G 10,00G
[root@sal]#
```

5.2.2. Ampliación de un grupo de volumen

Cuando nos quedamos sin espacio en un grupo de volumen siempre podemos ampliarlo utilizando más dispositivos físicos o particiones. Para ello utilizaremos el comando **vgextend**:

```
[root@sal]# vgextend data_vg /dev/sdd
 Volume group "data_vg" successfully extended
[root@sal]#
```

5.2.3. Reducción de un grupo de volumen

Podemos quitar dispositivos físicos de un grupo de volumen, siempre y cuando no esten en uso.

```
[root@sal]# vgreduce data_vg /dev/sdd
  Removed "/dev/sdd" from volume group "data_vg"
[root@sal]#
```

5.2.4. Activación y desactivación de grupos de volumen

Para poder utilizar los grupos de volumen es necesario que esten activos. Utilizaremos el comando **vgchange** para activarlos y desactivarlos.

---

sugerencia
El comando **vgchange** se utilizar para modificar los parámetros de los grupos de volumen.

5.2.5. Importación y exportación de grupos de volumen

Hay ocasiones en las que es necesario mover discos entre diferentes máquinas. Por ejemplo imaginemos que tenemos varias instancias de BBDD en una máquina.

La configuración ideal para un entorno de este tipo es que cada instancia tenga sus datos en su propia partición y cada partición este en un grupo de volumen diferente. De esta forma si alguna de las instancias crece, en términos de potencia, siempre es posible cambiar alguna instacia a otra máquina.

En este caso tendremos que exportar el grupo de volumen que queremos mover a otra máquina y posteriormente importarlo.

Será necesario desactivar todas las particiones lógicas de ese grupo de volumen:

```
[root@sal]# lvchange -a n /dev/data_vg/mysql_lv
[root@sal]# vgexport data_vg
 Volume group "data_vg" successfully exported
[root@sal]#
```

Ahora podemos mover los dispositivos físicos que forman el grupo de volumen *data_vg* a otra máquina e importarlos para poder utilizarlos:

```
[root@pal]# vgimport -a
  Volume group "data_vg" successfully imported
[root@pal]#
```

5.2.6. Eliminación de un grupo de volumen

Para poder eliminar un grupo de volumen necesitaremos que no esté en uso ninguna de las particiones lógicas presentes y que este desactivado.

```
[root@sal]# vgremove system_vg
  Volume group "system_vg" successfully removed
[root@sal]#
```

5.3. Particiones lógicas (logical volumes)

Las particiones lógicas son particiones que se crean dentro de un volume group y podrán crecer siempre y cuando hay espacio libre dentro del grupo de volumen.

Una partición lógica se crea:

```
[root@sal]# lvcreate -L5G -n mysql_lv /dev/data_vg
  Logical volume "mysql_lv" created
[root@sal]#
```

Habremos creado una partición lógica de 5 gigas en el grupo de volumen *data_vg*.

---

sugerencia
Para formatearla, montarla, ... nos referiremos a ella como `/dev/data_vg/mysql_lv`.

5.3.1. Información y detección de particiones lógicas

Podemos utilizar el comando **lvscan** para encontrar particiones lógicas:

```
[root@sal]# lvscan
  ACTIVE            '/dev/data_vg/apache_lv' [20,00 GB] inherit
  ACTIVE            '/dev/data_vg/mysql_lv' [20,00 GB] inherit
[root@sal]#
```

También disponemos del comando **lvdisplay** que nos ofrece más información sobre las particiones lógicas encontradas:

```
[root@sal]# lvdisplay
  --- Logical volume ---
  LV Name                /dev/data_vg/apache_lv
  VG Name                data_vg
  LV UUID                KCDc3t-jAHj-dNDK-I9qe-rF3B-oyya-Tg00JL
  LV Write Access        read/write
  LV Status              available
  # open                 1
  LV Size                20,00 GB
  Current LE             5120
  Segments               1
  Allocation             inherit
  Read ahead sectors     0
  Block device           253:0

  --- Logical volume ---
  LV Name                /dev/data_vg/mysql_lv
  VG Name                data_vg
  LV UUID                0m24gV-H9BC-MdbV-vSNZ-r57x-gx37-zUy22q
  LV Write Access        read/write
  LV Status              available
```

```
  # open                 0
  LV Size                20,00 GB
  Current LE             5120
  Segments               1
  Allocation             inherit
  Read ahead sectors     0
  Block device           253:1
[root@sal]#
```

El comando **lvs** también nos ofrece información:

```
[root@sal]# lvs
  LV        VG      Attr   LSize  Origin Snap%  Move Log Copy%
  apache_lv data_vg -wi-ao 20,00G
  mysql_lv  data_vg -wi-a- 20,00G
[root@sal]#
```

5.3.2. Ampliación de una partición lógica

```
[root@sal]# lvextend -L+2G /dev/data_vg/mysql_lv
  Extending logical volume mysql_lv to 7,00 GB
  Logical volume mysql_lv successfully resized
[root@sal]#
```

Una vez ampliada la partición lógica habrá que hacer un resize. Si es sistema de ficheros es *ReiserFS* podemos hacerlo en caliente sin desmontarlo:

```
[root@sal]# resize_reiserfs /dev/data_vg/mysql_lv
  resize_reiserfs 3.6.19 (2003 www.namesys.com)


  resize_reiserfs: On-line resize finished successfully.
[root@sal]#
```

Con *ext3*:

```
[root@sal]# ext2online /dev/data_vg/mysql_lv
ext2online v1.1.18 - 2001/03/18 for EXT2FS 0.5b
[root@sal]#
```

5.3.3. Reducción de tamaño para particiones lógicas

aviso
Antes de reducir un sistema de ficheros es MUY recomendable asegurarse de que hay un backup de los datos.

aviso
Si el sistema de ficheros está fragmentado y existen datos en la parte a reducir esos datos se perderán.

aviso
Cuando reduzcamos un sistema de ficheros tenemos que asegurarnos de que el sistema de ficheros resultante puede contener todos los datos.

Los pasos a seguir son:

- Desmontar la partición.
- Hacer un resize del sistema de ficheros al tamaño deseado.

  Con un sistema de ficheros *ReiserFS*:

```
[root@sal]# resize_reiserfs -s -1G /dev/data_vg/mysql_lv
Dando formato a resize_reiserfs(8); aguarde, por favor...
telemaco:/media# resize_reiserfs -s -1G /dev/data_vg/mysql_lv
resize_reiserfs 3.6.19 (2003 www.namesys.com)

You are running BETA version of reiserfs shrinker.
This version is only for testing or VERY CAREFUL use.
Backup of you data is recommended.

Do you want to continue? [y/N]:y
Processing the tree: 0%....20%....40%....60%....80%....100%                     left 0,  ↩
    105322 /sec

nodes processed (moved):
int        3 (0),
leaves     209 (0),
unfm       210432 (0),
total      210644 (0).

check for used blocks in truncated region

ReiserFS report:
blocksize             4096
block count           1048576 (1310720)
free blocks           829690 (1091826)
bitmap block count    32 (40)

Syncing..done


resize_reiserfs: Resizing finished successfully.
[root@sal]#
```

  Con un sistema de ficheros *ext3*:

```
[root@sal]# e2fsck -f /dev/data_vg/mysql_lv
e2fsck 1.35 (28-Feb-2004)
Paso 1: revisando nodos i, bloques y tamaños
Paso 2: revisando la estructura de directorios
Paso 3: revisando la conectividad del directorio.
Paso 4: revisando las cuentas de referencia
Paso 5: revisando el resumen de información del grupo
/dev/data_vg/mysql_lv: ficheros 11/92160 (9.1% no contiguos), bloques 7156/179200
[root@sal]# resize2fs /dev/data_vg/mysql_lv 6g
resize2fs 1.35 (28-Feb-2004)
Resizing the filesystem on /dev/data_vg/mysql_lv to 2816000 (4k) blocks.
El sistema de ficheros en /dev/data_vg/mysql_lv mide ahora 2816000 bloques.
```

- Reducimos la partición lógica:

```
[root@sal]# lvreduce -L-1G /dev/data_vg/mysql_lv
  WARNING: Reducing active logical volume to 6,00 GB
  THIS MAY DESTROY YOUR DATA (filesystem etc.)
Do you really want to reduce mysql_lv? [y/n] y
  Logical volume mysql_lv successfully resized
[root@sal]#
```

5.3.4. Activación y desactivación de particiones lógicas

Para determinadas operaciones de los grupos de volumen es necesario desactivar las particiones lógicas que en ellos residen.

Utilizaremos para ello el comando **lvchange**:

```
[root@sal]# lvchange -a n /dev/data_vg/mysql_lv
[root@sal]#
```

Desactivaría la partición lógica.

---

sugerencia
man lvchange

5.3.5. Eliminación de una partición lógica

Para eliminar una partición lógica utilizaremos el comando **lvremove**:

```
[root@sal]# lvremove /dev/data_vg/mysql_lv
Do you really want to remove active logical volume "mysql_lv"? [y/n]: y
  Logical volume "mysql_lv" successfully removed
[root@sal]#
```

Capítulo 6

Introducción al uso de *SAN* en GNU/Linux

Las redes *SAN* o *S*torage *A*rea *N*etwork están siendo cada vez más utilizadas debido a la potencia y escalabilidad que presentan.

El proposito de este capítulo es ofrecer una visión general del uso de almacenamiento externo en GNU/Linux y no el uso de software suministrado por cada fabricante para el manejo de sus dispositivos de almacenamiento.

6.1. Breve introducción a una *SAN*

Los discos ocupan espacio y cuando los requerimientos de disco crecen muchas veces no es posible añadir los discos a un servidor debido a problemas de espacio.

Para solucionar esto se ha recurrido al almacenmamiento externo.

Una *SAN* no es máte más que una cabina o armario de discos conectados por fibra óptica a los servidores.

Para esta conexión se utilizan unos switches especiales, de fibra, que son los que están conectados a las cabinas y a los servidores mediante tarjetas de fibra o *HBAs*.

Las configuraciones habituales son dos *HBAs* por máquina.

Los discos habitualmente se configuran para que se llegue por varios caminos por tarjeta. Normalmente dos caminos por tarjeta. Por este motivo cada disco se veráa por cuatro caminos. Es decir que tendremos cuatro dispositivos físicos que son el mismo.

Esto se denomina *multipathing* y permite el balanceo de carga y la alta disponibilidad en el acceso a disco.

Hay diferentes fabricantes que ofrencen soluciones *SAN*. Los más conocidos:

- *IBM*
- *EMC2*
- *Hitachi*
- *HP*

Para el uso del *multipathing* cada fabricante proporciona su propio software y será necesario utilizarlo si queremos disponer de las capacidades de *multipathing*.

---

sugerencia
Utilizando *device mapper* podemos hacer *multipathing*. Si nos decidimos a utilizar *device mapper* nos ahorraremos bastante dinero en concepto de licencias pero hay que tener en cuenta que no dispondremos de soporte oficial, aunque lo paguemos, por parte del fabricante de las cabinas.
El uso de software no certificado puede inducir problemas en la red *SAN* que pudieran afectar a otros equipos. Por este motivo lo recomendable es utilizar el software proporcionado por el fabricante.

6.2. Escaneado de discos

Cada vez que necesitemos disco es posible asignarle disco a un servidor y añadirselo sin reiniciarlo.

Para añadir los discos será necesario hacerlo de dos pasos:

1. Reescaneo del bus *SCSI*.

   En el caso de tarjetas *Qlogic 2340*:

   ```
   [root@sal]# echo "scsi-qlascan" > /proc/scsi/qla2300/1
   [root@sal]#
   ```

   Tendremos que hacer esto para cada tarjeta a la que se haya asignado el disco. Las tarjetas serán nombradas con un número dentro de `/proc/scsi/qla2300/`.

   Una vez hecho esto el sistema SCSI verá los discos pero será necesario registrarlos en el sistema para asignarles un dispositivo:

   ```
   [root@sal]# cat /proc/scsi/qla2300/1
   QLogic PCI to Fibre Channel Host Adapter for QLA2340:
           Firmware version:  3.02.13, Driver version 6.06.00
   Entry address = ce800060
   HBA: QLA2312 , Serial# P19322
   Request Queue = 0xe8ec000, Response Queue = 0xe8d0000
   Request Queue count= 128, Response Queue count= 512
   Total number of active commands = 0
   Total number of interrupts = 12
   Total number of IOCBs (used/max) = (0/600)
   Total number of queued commands = 0
       Device queue depth = 0x20
   Number of free request entries = 127
   Number of mailbox timeouts = 0
   Number of ISP aborts = 0
   Number of loop resyncs = 0
   Number of retries for empty slots = 0
   Number of reqs in pending_q= 0, retry_q= 0, done_q= 0, scsi_retry_q= 0
   Host adapter:loop state= <READY>, flags= 0x8e0813
   Dpc flags = 0x0
   MBX flags = 0x0
   SRB Free Count = 4096
   Link down Timeout = 000
   Port down retry = 030
   Login retry count = 030
   Commands retried with dropped frame(s) = 0


   SCSI Device Information:
   scsi-qla0-adapter-node=200000e08b17da2e;
   scsi-qla0-adapter-port=210000e08b17da2e;
   scsi-qla0-target-0=5005076300c4a585;
   scsi-qla0-target-1=5005076300c3a585;
   scsi-qla0-target-2=5005076300c2a585;
   scsi-qla0-target-3=5005076300cca585;
   scsi-qla0-target-4=5005076300cba585;
   scsi-qla0-target-5=5005076300caa585;

   SCSI LUN Information:
   (Id:Lun)  * - indicates lun is not registered with the OS.
   ( 0: 0): Total reqs 0, Pending reqs 0, flags 0x0*, 0:0:81,
   ( 0: 1): Total reqs 0, Pending reqs 0, flags 0x0*, 0:0:81,
   ( 0: 2): Total reqs 0, Pending reqs 0, flags 0x0*, 0:0:81,
   ( 1: 0): Total reqs 0, Pending reqs 0, flags 0x0*, 0:0:82,
   ```

```
( 1: 1): Total reqs 0, Pending reqs 0, flags 0x0*, 0:0:82,
( 1: 2): Total reqs 0, Pending reqs 0, flags 0x0*, 0:0:82,
( 2: 0): Total reqs 0, Pending reqs 0, flags 0x0*, 0:0:83,
( 2: 1): Total reqs 0, Pending reqs 0, flags 0x0*, 0:0:83,
( 2: 2): Total reqs 0, Pending reqs 0, flags 0x0*, 0:0:83,
( 3: 0): Total reqs 0, Pending reqs 0, flags 0x0*, 0:0:84,
( 3: 1): Total reqs 0, Pending reqs 0, flags 0x0*, 0:0:84,
( 3: 2): Total reqs 0, Pending reqs 0, flags 0x0*, 0:0:84,
( 4: 0): Total reqs 0, Pending reqs 0, flags 0x0*, 0:0:85,
( 4: 1): Total reqs 0, Pending reqs 0, flags 0x0*, 0:0:85,
( 4: 2): Total reqs 0, Pending reqs 0, flags 0x0*, 0:0:85,
( 5: 0): Total reqs 0, Pending reqs 0, flags 0x0*, 0:0:86,
( 5: 1): Total reqs 0, Pending reqs 0, flags 0x0*, 0:0:86,
( 5: 2): Total reqs 0, Pending reqs 0, flags 0x0*, 0:0:86,
[root@sal]#
```

Aquellos dispositivos en los que aparezca *flags 0x0**, el asterisco nos da la clave, son los dispositivos nuevos que se han asignado y para los que necesitaremos registrar en el sistema para asignarles un dispositivo en `/dev/`.

En el caso de tarjetas *Qlogic 2462*:

```
[root@sal]# echo 1 > /sys/class/fc_host/host1/issue_lip
[root@sal]#
```

Tendremos que hacer esto para cada tarjeta a la que se haya asignado el disco. Las tarjetas serán nombradas con host*n*.

Una vez hecho esto el sistema SCSI verá los discos pero será necesario registrarlos en el sistema para asignarles un dispositivo:

```
[root@sal]# cat /proc/scsi/qla2xxx/1
QLogic PCI to Fibre Channel Host Adapter for QMC2462S:
        Firmware version 4.00.18 [IP] , Driver version 8.01.04-d7
ISP: ISP2422
Request Queue = 0x7cc00000, Response Queue = 0x7d3c0000
Request Queue count = 4096, Response Queue count = 512
Total number of active commands = 28
Total number of interrupts = 15416672
    Device queue depth = 0x20
Number of free request entries = 4067
Number of mailbox timeouts = 0
Number of ISP aborts = 0
Number of loop resyncs = 0
Number of retries for empty slots = 0
Number of reqs in pending_q= 0, retry_q= 0, done_q= 0, scsi_retry_q= 0
Host adapter:loop state = <READY>, flags = 0x1e03
Dpc flags = 0x4000000
MBX flags = 0x0
Link down Timeout = 030
Port down retry = 030
Login retry count = 030
Commands retried with dropped frame(s) = 0
Product ID = 0000 0000 0000 0000


SCSI Device Information:
scsi-qla0-adapter-node=200000e08b859383;
scsi-qla0-adapter-port=210000e08b859383;
scsi-qla0-target-0=5006016030224a8b;
scsi-qla0-target-1=5006016930224a8b;

FC Port Information:
scsi-qla0-port-0=50060160b0224a8b:5006016030224a8b:010000:81;
scsi-qla0-port-1=50060160b0224a8b:5006016930224a8b:010400:82;
```

```
SCSI LUN Information:
(Id:Lun)  * - indicates lun is not registered with the OS.
( 0: 0): Total reqs 256853, Pending reqs 0, flags 0x0, 0:0:81 00
( 0: 1): Total reqs 8896695, Pending reqs 0, flags 0x0, 0:0:81 00
( 0: 2): Total reqs 8524762, Pending reqs 28, flags 0x0, 0:0:81 00
( 0: 3): Total reqs 216957, Pending reqs 0, flags 0x0, 0:0:81 00
( 0: 4): Total reqs 0, Pending reqs 0, flags 0x0*, 0:0:81 00
( 0: 5): Total reqs 0, Pending reqs 0, flags 0x0*, 0:0:81 00
( 0: 6): Total reqs 0, Pending reqs 0, flags 0x0*, 0:0:81 00
( 1: 0): Total reqs 14430, Pending reqs 0, flags 0x0, 0:0:82 00
( 1: 1): Total reqs 14442, Pending reqs 0, flags 0x0, 0:0:82 00
( 1: 2): Total reqs 14463, Pending reqs 0, flags 0x0, 0:0:82 00
( 1: 3): Total reqs 14399, Pending reqs 0, flags 0x0, 0:0:82 00
( 1: 4): Total reqs 0, Pending reqs 0, flags 0x0*, 0:0:82 00
( 1: 5): Total reqs 0, Pending reqs 0, flags 0x0*, 0:0:82 00
( 1: 6): Total reqs 0, Pending reqs 0, flags 0x0*, 0:0:82 00
[root@sal]#
```

Aquellos dispositivos en los que aparezca *flags 0x0**, el asterisco nos da la clave, son los dispositivos nuevos que se han asignado y para los que necesitaremos registrar en el sistema para asignarles un dispositivo en `/dev/`.

Tendremos que hacer esto para cada tarjeta a la que se haya asignado el disco. Las tarjetas serán nombradas como host*n*.

Una vez hecho esto el sistema SCSI verá los discos pero será necesario registrarlos en el sistema para asignarles un dispositivo.

2. Registro de los discos en el sistema. Esto le asignará a cada dispositivo un dispositivo físico en `/dev/`.

   En el caso de tarjetas *Qlogic 2340*:

   Por cada dispositivo que presente un "*" en cada tarjeta tendremos que hacer:

```
[root@sal]# echo "add-single-device  R C T L" > /proc/scsi/scsi
[root@sal]#
```

   Donde:

   - *R* es la tarjeta. El número dentro de `/proc/scsi/qla2300/`.
   - *C* es el canal. Normalmente es cero. Se puede verificar en `/proc/scsi/scsi`.
   - *T* es el target y viene especificado por el campo *Id*.
   - *L* es el lun.

   Una vez registrados todos los dispositivos:

```
[root@sal]# cat /proc/scsi/qla2300/1
QLogic PCI to Fibre Channel Host Adapter for QLA2340:
        Firmware version:  3.02.13, Driver version 6.06.00
Entry address = ce800060
HBA: QLA2312 , Serial# P19322
Request Queue = 0xe8ec000, Response Queue = 0xe8d0000
Request Queue count= 128, Response Queue count= 512
Total number of active commands = 0
Total number of interrupts = 12
Total number of IOCBs (used/max) = (0/600)
Total number of queued commands = 0
    Device queue depth = 0x20
Number of free request entries = 127
Number of mailbox timeouts = 0
Number of ISP aborts = 0
Number of loop resyncs = 0
Number of retries for empty slots = 0
Number of reqs in pending_q= 0, retry_q= 0, done_q= 0, scsi_retry_q= 0
```

```
Host adapter:loop state= <READY>, flags= 0x8e0813
Dpc flags = 0x0
MBX flags = 0x0
SRB Free Count = 4096
Link down Timeout = 000
Port down retry = 030
Login retry count = 030
Commands retried with dropped frame(s) = 0


SCSI Device Information:
scsi-qla0-adapter-node=200000e08b17da2e;
scsi-qla0-adapter-port=210000e08b17da2e;
scsi-qla0-target-0=5005076300c4a585;
scsi-qla0-target-1=5005076300c3a585;
scsi-qla0-target-2=5005076300c2a585;
scsi-qla0-target-3=5005076300cca585;
scsi-qla0-target-4=5005076300cba585;
scsi-qla0-target-5=5005076300caa585;

SCSI LUN Information:
(Id:Lun)  * - indicates lun is not registered with the OS.
( 0: 0): Total reqs 0, Pending reqs 0, flags 0x0, 0:0:81,
( 0: 1): Total reqs 0, Pending reqs 0, flags 0x0, 0:0:81,
( 0: 2): Total reqs 0, Pending reqs 0, flags 0x0, 0:0:81,
( 1: 0): Total reqs 0, Pending reqs 0, flags 0x0, 0:0:82,
( 1: 1): Total reqs 0, Pending reqs 0, flags 0x0, 0:0:82,
( 1: 2): Total reqs 0, Pending reqs 0, flags 0x0, 0:0:82,
( 2: 0): Total reqs 0, Pending reqs 0, flags 0x0, 0:0:83,
( 2: 1): Total reqs 0, Pending reqs 0, flags 0x0, 0:0:83,
( 2: 2): Total reqs 0, Pending reqs 0, flags 0x0, 0:0:83,
( 3: 0): Total reqs 0, Pending reqs 0, flags 0x0, 0:0:84,
( 3: 1): Total reqs 0, Pending reqs 0, flags 0x0, 0:0:84,
( 3: 2): Total reqs 0, Pending reqs 0, flags 0x0, 0:0:84,
( 4: 0): Total reqs 0, Pending reqs 0, flags 0x0, 0:0:85,
( 4: 1): Total reqs 0, Pending reqs 0, flags 0x0, 0:0:85,
( 4: 2): Total reqs 0, Pending reqs 0, flags 0x0, 0:0:85,
( 5: 0): Total reqs 0, Pending reqs 0, flags 0x0, 0:0:86,
( 5: 1): Total reqs 0, Pending reqs 0, flags 0x0, 0:0:86,
( 5: 2): Total reqs 0, Pending reqs 0, flags 0x0, 0:0:86,
[root@sal]#
```

En el caso de tarjetas *Qlogic 2462*:

Por cada dispositivo que presente un "*" en cada tarjeta tendremos que hacer:

```
[root@sal]# echo "- - -" > /sys/class/scsi_host/host1/scan
[root@sal]#
```

importante
Cada una de las "-" hace referencia a *bus*, *target* y *lun* del dispositivo.

Con esto registramos todos los dispositivos físicos que están sin registrar en el sistema para la tarjeta *host1*:

```
[root@sal]# cat /proc/scsi/qla2xxx/1
QLogic PCI to Fibre Channel Host Adapter for QMC2462S:
        Firmware version 4.00.18 [IP] , Driver version 8.01.04-d7
ISP: ISP2422
```

```
Request Queue = 0x7cc00000, Response Queue = 0x7d3c0000
Request Queue count = 4096, Response Queue count = 512
Total number of active commands = 0
Total number of interrupts = 15422987
    Device queue depth = 0x20
Number of free request entries = 2201
Number of mailbox timeouts = 0
Number of ISP aborts = 0
Number of loop resyncs = 0
Number of retries for empty slots = 0
Number of reqs in pending_q= 0, retry_q= 0, done_q= 0, scsi_retry_q= 0
Host adapter:loop state = <READY>, flags = 0x1e03
Dpc flags = 0x4000000
MBX flags = 0x0
Link down Timeout = 030
Port down retry = 030
Login retry count = 030
Commands retried with dropped frame(s) = 0
Product ID = 0000 0000 0000 0000


SCSI Device Information:
scsi-qla0-adapter-node=200000e08b859383;
scsi-qla0-adapter-port=210000e08b859383;
scsi-qla0-target-0=5006016030224a8b;
scsi-qla0-target-1=5006016930224a8b;

FC Port Information:
scsi-qla0-port-0=50060160b0224a8b:5006016030224a8b:010000:81;
scsi-qla0-port-1=50060160b0224a8b:5006016930224a8b:010400:82;

SCSI LUN Information:
(Id:Lun)  * - indicates lun is not registered with the OS.
( 0: 0): Total reqs 256859, Pending reqs 0, flags 0x0, 0:0:81 00
( 0: 1): Total reqs 8897026, Pending reqs 0, flags 0x0, 0:0:81 00
( 0: 2): Total reqs 8533818, Pending reqs 0, flags 0x0, 0:0:81 00
( 0: 3): Total reqs 216979, Pending reqs 0, flags 0x0, 0:0:81 00
( 0: 4): Total reqs 76, Pending reqs 0, flags 0x0, 0:0:81 00
( 0: 5): Total reqs 77, Pending reqs 0, flags 0x0, 0:0:81 00
( 0: 6): Total reqs 77, Pending reqs 0, flags 0x0, 0:0:81 00
( 1: 0): Total reqs 14431, Pending reqs 0, flags 0x0, 0:0:82 00
( 1: 1): Total reqs 14444, Pending reqs 0, flags 0x0, 0:0:82 00
( 1: 2): Total reqs 14463, Pending reqs 0, flags 0x0, 0:0:82 00
( 1: 3): Total reqs 14450, Pending reqs 0, flags 0x0, 0:0:82 00
( 1: 4): Total reqs 15, Pending reqs 0, flags 0x0, 0:0:82 00
( 1: 5): Total reqs 15, Pending reqs 0, flags 0x0, 0:0:82 00
( 1: 6): Total reqs 21, Pending reqs 0, flags 0x0, 0:0:82 00
[root@sal]#
```

importante
Es necesario hacer esto para cada tarjeta, *hostn*, que tenga dispositivos sin registrar en el sistema.

6.3. Dispositivos virtuales

Para utilizar las capacidades de multipathing es necesario utilizar el dispositivo virtual que crea el software proporcionado por el fabricante:

- Los dispositivos virtuales utilizados por el driver multipath de *IBM*, *SDD* (*S*ubsystem *D*evice *D*river), son `/dev/vpatha`, `/dev/vpathb`, ...
- Los dispositivos virtuales utilizados por el driver multipath de *EMC*, *Powerpath* son `/dev/emcpowera`, `/dev/emcpowerb`, ...

6.4. Multipathing utilizando *LVM*

Es posible hacer multipathing utilizando *LVM* pero sólo con unas determinadas versiones. Existen parches para las versiones *1.0.5*, *1.0.6*, *1.0.7* y *1.0.8*.

La única distribución en la que viene configurado el kernel para el uso de este multipathing es *SLES 8*.

6.4.1. Localizando los dispositivos físicos

Una vez añadidos los dispositivos físicos al sistema tendremos tantos dispositivos como caminos por disco. Para cada disco físico utilizaremos un dispositivo especifico de todos los que lo referencian, lo llamaremos *dispositivo primario* para ese disco. El comando *pvscan* nos identificará estos dispositivos:

```
[root@sal]# pvscan
pvscan -- reading all physical volumes (this may take a while...)
pvscan -- ACTIVE   PV "/dev/sdaw"  of VG "data_vg"     [18.62 GB / 0 free]
pvscan -- ACTIVE   PV "/dev/sdax"  of VG "data_vg"     [46.56 GB / 26.98 GB free]
pvscan -- ACTIVE   PV "/dev/sdav"  of VG "data_vg"     [46.56 GB / 46.56 GB free]
pvscan -- ACTIVE   PV "/dev/sdag"  of VG "data_vg"     [46.56 GB / 0 free]
pvscan -- ACTIVE   PV "/dev/sdah"  of VG "data_vg"     [9.31 GB / 0 free]
pvscan -- ACTIVE   PV "/dev/sdai"  of VG "data_vg"     [46.56 GB / 0 free]
pvscan -- ACTIVE   PV "/dev/sdaj"  of VG "data_vg"     [46.56 GB / 0 free]
pvscan -- ACTIVE   PV "/dev/sdak"  of VG "data_vg"     [19.18 GB / 0 free]
pvscan -- ACTIVE   PV "/dev/sdal1" of VG "software_vg" [5 GB / 116 MB free]
pvscan -- ACTIVE   PV "/dev/sdal2" of VG "data_vg"     [14.18 GB / 0 free]
pvscan -- ACTIVE   PV "/dev/sdad"  of VG "data_vg"     [46.56 GB / 0 free]
pvscan -- ACTIVE   PV "/dev/sdae"  of VG "data_vg"     [18.62 GB / 0 free]
pvscan -- ACTIVE   PV "/dev/sdaf"  of VG "data_vg"     [9.31 GB / 0 free]
pvscan -- ACTIVE   PV "/dev/sda13" of VG "system_vg"   [16.77 GB / 2.96 GB free]
pvscan -- total: 14 [390.43 GB] / in use: 14 [390.43 GB] / in no VG: 0 [0]
[root@sal]#
```

importante
Es necesario que los discos ya esten asignados a un grupo de volumen.

6.4.2. Configurando el multipath

Lo haremos en tres pasos:

1. Tendremos que configurar el multipath sobre los dispositivos primarios mostrados por **pvscan** suponiendo que hay cuatro caminos por dispositivo:

```
[root@sal]# pvpath -p 0 -e y -w 1 /dev/sdaw
[root@sal]# pvpath -p 1 -e y -w 2 /dev/sdaw
[root@sal]# pvpath -p 2 -e y -w 1 /dev/sdaw
[root@sal]# pvpath -p 3 -e y -w 2 /dev/sdaw
[root@sal]# pvpath -q /dev/sdaw
```

```
Physical volume /dev/sdaw of data_vg has 4 paths:
       Device  Weight Failed Pending State
 #  0:   8:16       1      0       0 enabled
 #  1:   8:32       2      0       0 enabled
 #  2:   8:48       1      0       0 enabled
 #  3:   8:64       2      0       0 enabled
[root@sal]#
```

2. Una vez configurados todos los caminos tendremos que grabar la configuración:

```
[root@sal]# pvpathsave
[root@sal]#
```

   Esta configuración se almacena en el fichero `/etc/pvpath.cfg`.

3. Tenemos que hacer que está configuración se lea antés de utilizar los dispositivos. Para ello nos aseguraremos de que el fichero `/etc/init.d/boot.local` contiene `/sbin/pvpathrestore`.

importante
En otras distribuciones diferentes de *SLES 8* será en un fichero equivalente.

importante
Si la máquina está arrancando desde *SAN* será necesario hacer este último paso si queremos que la máquina arranque.

Capítulo 7

Núcleo de Linux

El núcleo (kernel) es el "programa principal" de todo sistema operativo, a través de él se controla todo el sistema. El núcleo de Linux de tipo monolítico y esta basado en los núcleos de Unix.

Algunas características destacables son:

- Multitarea y multihilo, es capaz de ejecutar varios proceso al mismo tiempo, en incluso varios hilos. Soportando varios procesadores.
- El kernel y las aplicaciones corren en distintos espacios de trabajo. En el *kernel mode* se tiene acceso al hardware de la máquina e interrupciones, y en *user mode*, donde corren las aplicaciones, que tienen que acceder al hardware a través del kernel.
- Gestión de memoria a través del núcleo y memoria virtual para ampliar la memoria física disponible mediante la utilización de parte de sistema de ficheros como memoria.
- Soporte de librerías compartidas con las llamadas al sistema.
- Portabilidad, Linux esta disponible en muchas plataformas, desde grandes ordenadores hasta pdas.

Podemos ver la versión de nuestro núcleo actual con:

```
[pcm@sal]# uname -a
Linux merc 2.4.24-20040430 #1 SMP Fri Apr 30 21:34:00 CEST 2004 i686 GNU/Linux
[pcm@sal]#
```

7.1. Historia

En 1991 Linus Torvalds publicó en unas news de minix un nuevo kernel muy básico para procesadores Intel 386 y 486. Utilizando el compilador de C de GNU gcc y portando la shell bash.

La gente aportó código a ese núcleo inicial y así se fue ampliando y soportando más hardware. Se adoptó la licencia GNU GPL.

En el 1992 se creo ya su propio foro y se porto las X11. En 1994 apareció la versión 1.0. La versión 2 se comenzó en el 1996.

- La versión 2.2 comenzó en 1999 con 1.800.847 de lineas de código.
- La versión 2.4 comenzó en 2001 con 3.377.902 de lineas de código.
- La versión 2.6 comenzó en 2003 con 5.929.913 de lineas de código.
- La versiones impares (1.1, 2.3 o 2.5) son versiones de desarrollo, no son estables.
- Actualmente, a Abril del 2008, se acaba de publicar la versión 2.6.25.

7.2. Configurando un nuevo núcleo

Cuando arrancamos Linux ya tenemos configurado un núcleo de sistema, pero este puede que no contenga soporte para algún dispositivo o funcionalidad que queramos tener. También podemos querer optimizar o actualizar nuestro sistema.

Normalmente la distribución que estemos utilizando nos tendrá preparado y actualizado nuevas actualizaciones ya compiladas del núcleo, pero puede que no sea suficiente para lo que necesitamos.

Entonces tendremos que coger el código fuente del kernel, configurarlo, compilarlo nosotros e instalarlo.

7.2.1. Obtener los fuentes del núcleo

Normalmente las distribuciones de Linux también nos proporcionan las fuentes de núcleo como paquete. Suelen ser las versiones estables. Si queremos una versión más moderna o de desarrollo podemos directamente descargarnos los fuentes de www.kernel.org.

Una vez instalado el paquete o descargado descomprimimos, deberá colocarse en `/usr/src`.

7.2.2. Configuración

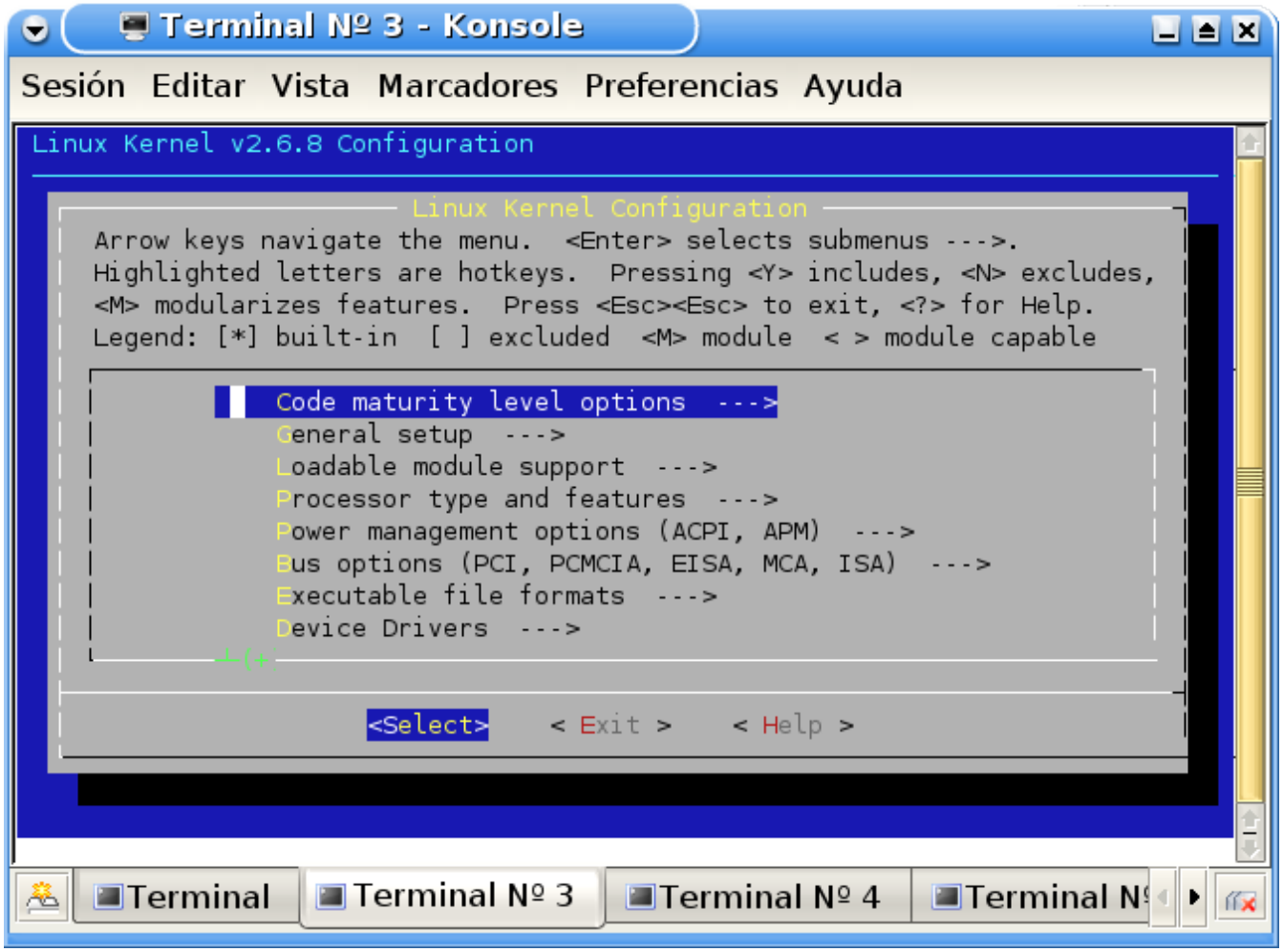
Para configurar los fuentes del núcleo debemos ir a directorio donde se encuentran y ejecutar **make** con una de las siguientes opciones.

- *config*: nos irá preguntando por consola una por una las opciones del núcleo. Debiendo responder si la queremos o no.

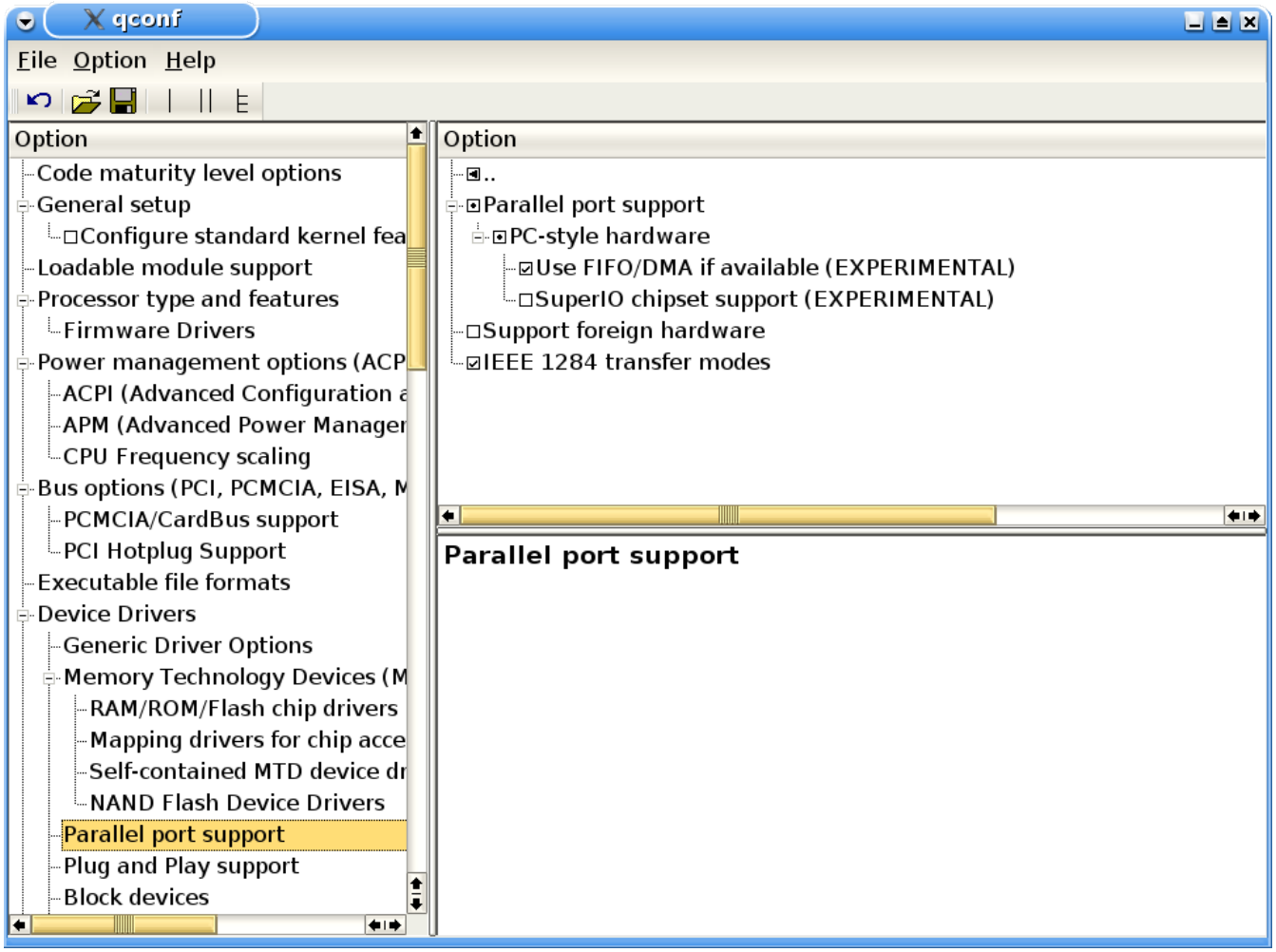
```
[root@sal]# make config

scripts/kconfig/conf arch/i386/Kconfig
#
# using defaults found in .config
#
*
* Linux Kernel Configuration
*
*
* Code maturity level options
*
Prompt for development and/or incomplete code/drivers (EXPERIMENTAL) [Y/n/?]
y
```

- *menuconfig*: podremos configurar las opciones con una pantalla con menús textuales, moviendo con los cursores entrando en las opciones y seleccionado las opciones.



- *xconfig*: Nos permite utilizar en entorno X Windows.



Con estas presentaciones de opciones se pretende que vayamos seleccionado que dispositivos, protocolos, sistemas de ficheros y otras opciones del núcleo queremos que estén disponibles en nuestro nuevo núcleo. Se nos presentan las opciones ordenadas por grupos:

- *Code maturity level options*: Contiene opciones para que nos muestre o no opciones del núcleo que todavía no están suficientemente probadas. Sobre todo son drivers para nuevos dispositivos.
- *General setup*: Opciones generales, sobre si el núcleo va a tener llamadas de sistema de estándares de UNIX.
- *Loadable module support*: Soporte de módulos en el núcleo. Se abordará en una sección posterior.
- *Processor type and features*: Nos permite compilar el núcleo más específicamente para procesador.
- *Power management options (ACPI, APM)*: Soporte para los sistemas de ahorro de energía.
- *Bus options (PCI, PCMCIA, EISA, MCA, ISA)*: Soporte para los distintos buses de datos.
- *Executable file formats*: Los distintos formatos de binarios en que pueden encontrarse en las aplicaciones, que el núcleo reconocerá y será capaz de ejecutar.
- *Device Drivers*: Opciones para todos los dispositivos hardware que Linux soporta. Este grupo es muy extenso y esta divido por los tipos de hardware. Suele ser el motivo por el cual recompilamos el núcleo.
- *File systems*: Tipos de sistemas de ficheros que nos soporta el núcleo
- *Profiling support*: Opciones que nos van a permitir analizar el comportamiento del núcleo. Esto se hace normalmente para desarrollo y optimización.

- *Kernel hacking*: Soporte para depurar el núcleo.
- *Security options*: Opciones para sistemas de seguridad que permiten que incluso el superusuario no tenga totalmente el control de la máquina.
- *Cryptographic options*: API para que el núcleo tenga opciones de criptografía. Algunos drivers lo utilizan, por ejemplo los driver para tarjetas y redes wifi.
- *Library routines*: Librerias disponibles para el resto de driver y opciones.

Siempre para cada opción o grupo tenemos una pequeña ayuda que nos indica para que sirve y aconseja que opción es la mejor.

Una vez seleccionadas podemos guardar la configuración en un fichero a parte o bien al salir nos preguntará si queremos guardarlo en el fichero `.config` que será el utilizado en la compilación.

7.2.3. Compilando el núcleo

Una vez configurada las opciones y guardadas las opciones en el fichero `.config` debemos compilar para ello ejecutaremos el comando **make** con las siguientes opciones:

```
[root@sal]# make dep
...

[root@sal]# make bzImage
...

[root@sal]#
```

Nos creará el bloque principal del núcleo. Antes de instalarlo debermos crear los modulos

---

sugerencia
Debian dispone unas herramientas en paquete `kernel-package`, que nos permiten compilar y crear un paquete *deb* con el núcleo compilado ya preparado para su instalación. Haciendo mucho más sencillo el proceso y facilitando la instalación en otros sistemas con esta distribución.

7.2.4. Módulos de núcleo

Aunque se supone que el núcleo es monolítico, es decir, todas los servicios del núcleo están en un solo bloque, realmente en Linux se pueden cargar dinámicamente módulos.

Esto nos permite durante la ejecución cargar solo los drivers para los dispositivos que estemos utilizando. También hace que el programa principal sea más pequeño y ocupe menos en memoria y sólo se ocupe la memoria cuando estemos utilizando el módulo.

Las distribuciones lo que suelen hacer para tener un núcleo que tenga soporte para cualquier sistema es compilar la mayor parte de las opciones como núcleo, así es poco probable que todo nuestro hardware no este soportado. Aunque se supone que es más óptimo que un driver este en el núcleo principal en vez de como módulo.

Para crear los módulos debemos en el momento de la configuración del núcleo establecer que queremos soporte para módulos, y poner en la configuración las opciones que queremos como módulos. Para ello al establecer una opción como módulo con los distintos sistemas de configuración (config, menuconfig y xconfig) debemos poner una M.

Para compilarlos debemos hacer **make modules**.

Una vez compilados cada módulo es un archivo binario con extensión `.o` en las versiones 2.4 y anteriores o `.ko` en las versiones 2.6. Los módulos instalados se encuentra en la ruta `/lib/modules/`*versión del núcleo*.

7.2.5. Instalando el núcleo

Una vez que tenemos el binario del núcleo y los módulos tenemos que instalarlo. Para ello tenemos que hacer **make modules_install** y **make install**.

7.2.6. Gestor de Arranque para el núcleo

Para que el nuevo kernel funcione debemos arrancar el sistema, para ello el gestor de arranque debe estar configurado y reinstalado con el nuevo núcleo.

El gestor de arranque es un pequeño programa instalado en los sectores de arranque que se encarga de arrancar Linux.

Existe dos gestores de arranque más conocidos (para plataformas intel), dependiendo de la distribución:

- *lilo*: Es el gestor más antiguo pero más probado.

  Para el nuevo núcleo debemos configurar su fichero de configuración `/etc/lilo.conf` con el nuevo núcleo en la etiqueta *image* y ejecutar **lilo**.

- *grub*: El gestor nuevo, gráficamente más atractivo y otras mejoras.

  El fichero de configuración es grub.conf, donde metemos el nuevo la localización del fichero binario del núcleo en la etiqueta *kernel*. Después ejecutamos el comando **grub**.

7.3. Configuración de parámetros del núcleo

El comando **sysctl** nos permite parametrizar parámetros del núcleo. No hace falta decir que son aquellos con los que fue compilado el núcleo.

Toda la configuración se hace a través del fichero `/etc/sysctl.conf`:

```
# Kernel sysctl configuration file
#
# For binary values, 0 is disabled, 1 is enabled.  See sysctl(8) and
# sysctl.conf(5) for more details.

# Controls IP packet forwarding
net.ipv4.ip_forward = 0

# Controls source route verification
net.ipv4.conf.default.rp_filter = 1

# Do not accept source routing
net.ipv4.conf.default.accept_source_route = 0

# Controls the System Request debugging functionality of the kernel
kernel.sysrq = 0

# Controls whether core dumps will append the PID to the core filename.
# Useful for debugging multi-threaded applications.
kernel.core_uses_pid = 1

net.ipv4.tcp_syncookies = 1
net.ipv4.tcp_fin_timeout = 60
net.ipv4.tcp_keepalive_time = 1800
net.ipv4.conf.all.accept_redirects = 0
net.ipv4.conf.all.send_redirects = 0
net.ipv4.conf.all.accept_source_route = 0
net.ipv4.conf.all.forwarding = 0
net.ipv4.conf.all.rp_filter = 1
net.ipv4.icmp_ignore_bogus_error_responses = 1
```

Esta configuración se activará en el arranque de la máquina o bien podemos modificar el fichero y:

```
[root@sal]# sysctl -p
net.ipv4.ip_forward = 0
net.ipv4.conf.default.rp_filter = 1
net.ipv4.conf.default.accept_source_route = 0
kernel.sysrq = 0
kernel.core_uses_pid = 1
net.ipv4.tcp_syncookies = 1
net.ipv4.tcp_fin_timeout = 60
net.ipv4.tcp_keepalive_time = 1800
net.ipv4.conf.all.accept_redirects = 0
net.ipv4.conf.all.send_redirects = 0
net.ipv4.conf.all.accept_source_route = 0
net.ipv4.conf.all.forwarding = 0
net.ipv4.conf.all.rp_filter = 1
net.ipv4.icmp_ignore_bogus_error_responses = 1
[root@sal]#
```

7.3.1. Modificación de los parámetros

Como ya hemos visto lo podemos hacer modificando el fichero de configuración `/etc/sysctl.conf` y ejecutando después **sysctl -p**.

También es posible modificar estos parámetros directamente en el sistema de ficheros `/proc/`.

Todos los parámetros que podemos configurar los encontraremos en `/proc/sys/`.

importante
Para algunas familias de dispositivos podemos establecer configuraciones por defecto, para todos, o por dispositivo individual. Por ejemplo para los dispositivos de red lo podemos hacer en */proc/sys/net/ipv4/conf/default/*, */proc/sys/net/ipv4/conf/all/eth?/*, */proc/sys/net/ipv4/conf/lo/*, ...

Por ejemplo el parámetro *net.ipv4.ip_forward* lo encontraremos en `/proc/sys/net/ipv4/ip_forward`. Para modificar este parámetro:

```
[root@sal]# echo 0 > /proc/sys/net/ipv4/ip_forward
[root@sal]#
```

El inconveniente de este método es que al reiniciar la máquina esta configuración se pierde.

Por lo tanto será necesario ejecutar los *echoes* y habrá que incluirlos en los ficheros de arranque.

El uso de *echoes* es una mala practica de administración y debe evitarse y usarse en su lugar el fichero `/etc/sysctl.conf`.

7.3.2. Parámetros configurables

Los parámetros que podremos configurar son aquellos que se hayan seleccionado en la configuración del núcleo que se está usando.

Podemos comprobar los parámetros que podemos configurar:

```
[root@sal]# sysctl -A

sunrpc.tcp_slot_table_entries = 16
sunrpc.udp_slot_table_entries = 16
sunrpc.max_resvport = 1023
sunrpc.min_resvport = 650
```

```
sunrpc.nlm_debug = 0
sunrpc.nfsd_debug = 0
sunrpc.nfs_debug = 0
sunrpc.rpc_debug = 0
abi.vsyscall32 = 1
dev.scsi.logging_level = 0
dev.raid.speed_limit_max = 200000
dev.raid.speed_limit_min = 1000
dev.cdrom.check_media = 0
dev.cdrom.lock = 1
dev.cdrom.debug = 0
dev.cdrom.autoeject = 0
dev.cdrom.autoclose = 1
dev.cdrom.info = CD-ROM information, Id: cdrom.c 3.20 2003/12/17
dev.cdrom.info =
dev.cdrom.info = drive name:            sr0
dev.cdrom.info = drive speed:           24
dev.cdrom.info = drive # of slots:      1
dev.cdrom.info = Can close tray:                1
dev.cdrom.info = Can open tray:         1
dev.cdrom.info = Can lock tray:         1
dev.cdrom.info = Can change speed:      1
dev.cdrom.info = Can select disk:       0
dev.cdrom.info = Can read multisession: 0
dev.cdrom.info = Can read MCN:          1
dev.cdrom.info = Reports media changed: 1
dev.cdrom.info = Can play audio:                1
dev.cdrom.info = Can write CD-R:                0
dev.cdrom.info = Can write CD-RW:       0
dev.cdrom.info = Can read DVD:          1
dev.cdrom.info = Can write DVD-R:       0
dev.cdrom.info = Can write DVD-RAM:     0
dev.cdrom.info = Can read MRW:          1
dev.cdrom.info = Can write MRW:         1
dev.cdrom.info = Can write RAM:         1
dev.cdrom.info =
dev.cdrom.info =
dev.rtc.max-user-freq = 64
debug.exception-trace = 1
net.ipv6.conf.default.max_addresses = 16
net.ipv6.conf.default.max_desync_factor = 600
net.ipv6.conf.default.regen_max_retry = 5
...
```

7.3.3. Algunos parámetros útiles

- *net.ipv4.tcp_syncookies = 1* proteje contra los ataques *Sync packet flooding* pero no es conforme a los estándares marcados por los RFCs.

aviso
Puede tener impacto en el rendimiento de servidores sobrecargados.

importante
Sync packet flooding es una ataque de denegación de servicio. Cuando se va a establecer una conexión TCP un host manda un paquete *SYNC* a lo que el otro host responde con un paquete *ACK*. Si el host atacante empieza a mandar de forma indiscriminada paquetes *SYNC* con la cabecera falseada con una IP de origen falsa el host atacado mandará sus *ACK* o bien a ningún host o a un host que no es el que está intentando establecer la comunicación. Dependiendo de la implementación de la pila TCP de ese host tratará el paquete *ACK* de una forma u otra.
El host atacado llegará un momento en el que no pueda atender más conexiones.
Otra forma de prevenir estos ataques es ampliando el tamaño de las colas en las que se almacena la informació referente a las peticiones de conexión: *net.ipv4.tcp_max_syn_backlog*.

- *net.ipv4.tcp_max_sync_backlog = XXX* por defecto suele estar a 1024 y estable el tamaño de la cola donde se guarda la información referente a las peticiones de conexión.
- *net.ipv4.tcp_fin_timeout = 60* cierra los sockets ináctivos, no se recivió paquete *FIN*, después de 60 segundos.
- *net.ipv4.tcp_keepalive_time = 1800* segundos después de un proceso de inactividad tras los cuales se intenta verificar si el cliente sigue vivo.
- *net.ipv4.conf.all.accept_redirects = 0* no se aceptan redirecciones ICMP para evitar ataques *Man in the middle*.

aviso
Mediante el uso del protocolo *ICMP* es posible modificar los gateways definidos estáticamente en la máquina.

- *net.ipv4.conf.all.send_redirects = 0* evita que la máquina mande paquetes *ICMP* con redirecciones.
- *net.ipv4.conf.all.accept_source_route = 0* se evita que TCP tenga control para decidir sobre la determinación de la ruta de los paquetes.
- *net.ipv4.conf.all.rp_filter = x* protección contra ataques de spoofing.

  x puede tomar los valores:

  - *0* no realiza comprobaciones.
  - *1* rechazar suplantaciones evidentes.
  - *2* comprobación exahustiva.
- *net.ipv4.ip_forward = 0* deshabilitar el reenvio de paquetes.

importante
Esta opción suele estar con un valor distinto de cero en máquinas que hacen enrutado entre diferentes redes.

- *net.ipv4.icmp_ignore_bogus_error_responses = 1* activa la protección ante mensajes de error malformados.
- *net.ipv4.icmp_echo_ignore_broadcast = 1* desactivar la respuestas a las peticiones de broadcast de echo ICMP.
- *net.ipv4.icmp_echo_ignore_all = 1* desactivació de la respuesta a ping.
- *kernel.panic = n* después de un *kernel panic* espera *n* segundos para reinciar el sistema.
- *kernel.sysrq = 1* activa *SYSRQ*.

  SYSRQ es una característica del núcleo que permite pasarle instrucciones con fines de depuración.

sugerencia
Con el *SYSRQ* activado si se presionan las teclas *AltGr+PetSis+t*, *AltGr+PetSis+m* y *AltGr+PetSis+p* se escribirán en los logs el estado de los procesos, memoria y CPU.

sugerencia
En la documentación del núcleo podemos encontrar toda la información sobre *sysctl* en `Documentacion/networking/ip-sysctl.txt`.

Capítulo 8

Usuarios y permisos en GNU/Linux

GNU/Linux es un sistema multiusuario y multitarea. Por este motivo en el sistema tienen que convivir diferentes usuarios y compartir los recursos del sistema.

Cada usuario tiene sus archivos donde guarda sus datos, trabajo, música, ... y necesita para ello mecanismos de seguridad que eviten que sus datos sean borrados, modificados o leídos por otros usuarios.

8.1. El *superusuario* o *root*

En los sistemas *UNIX* existe un usuario especial que es el encargado de poner orden entre el resto de usuarios. Este usuario recibe el nombre de *root* y tiene acceso a la totalidad del sistema.

importante
El usuario *root* es el encargado de realizar o delegar todas las tareas de mantenimiento y/o administración del sistema.

Los usuarios normales no tienen privilegios para cambiar las configuraciones del sistema o las aplicaciones a nivel global.

importante
Hay aplicaciones que permiten configuraciones personales a los usuarios. Estas configuraciones son especificas para cada usuario y no afectan al resto.

aviso
La cuenta de *root* no se suele utilizar salvo que sea absolutamente necesario. Al tener acceso ilimitado este usuario al sistema puede borrar datos o dejar al sistema inestable si se ejecuta el comando erroneo.

8.2. Grupos de usuarios

Los usuarios se agrupan en grupos. Un grupo no es más que un conjunto de usuarios con una tarea en común.

El fichero `/etc/group` contiene los grupos del sistema. Dentro de este fichero podremos ver grupos:

- *apache* grupo para usuarios o demonios que van a administrar el servidor web.
- *mysql* idem para el servidor de BBDD *MySQL*.

8.2.1. El fichero `/etc/group`

En este fichero se encuentran todos los grupos presentes en el sistema. Las entradas serán del tipo:

```
users:x:100:tux,pepito,pcm
```

Este fichero está compuesto por varias líneas con campos separados por ":":

- El primer campo es el nombre del grupo.
- El segundo campo es el password para el grupo. Normalmente no se utiliza y este campo contiene una "x".
- En el tercer campo tenemos el *GID* del grupo. GNU/Linux utiliza el GID para manejar los grupos. El nombre unicamente se utiliza para hacerle la vida más comoda al usuario.
- El cuarto campo son los usuarios que pertenecen a dicho grupo separados por comas.

aviso
No siempre aparecen en `/etc/group` los usuarios en el grupo al que pertenecen. Si no se han creado los usuarios de forma correcta no aparecerán.

Si listamos un fichero cualquiera:

```
-rw-r--r-- 1 jose users 1764 2007-04-08 18:38 admlinux.xml
```

Podemos ver que el fichero pertenece al usuario *jose* y al grupo *users* además de otra información.

importante
Es importante que el fichero `/etc/group` tenga permisos de lectura para todo el mundo ya que es a través de este fichero que se hace la conversión del GID al nombre del grupo. Si se quitara el permiso de lectura tendríamos algo como esto:

```
[pcm@sal]$ ls -l admlinux.xml
-rw-r--r-- 1 jose 100 1764 2007-04-08 18:38 admlinux.xml
[pcm@sal]$
```

Vemos que en la salida ya no aparece *users*. En su lugar aparece *100* que es el GID del grupo *users*.

8.2.2. Añdiendo grupos al sistema

Sólo el usuario *root* puede añdir grupos al sistema. Aunque puede conceder privilegios a otros para hacerlo.

Para añdir grupos al sistema se utiliza el comando **groupadd**:

```
[pcm@sal]# groupadd alumnos
[pcm@sal]#
```

Esto añadiría el grupo *alumnos* al sistema.

sugerencia
Si quisieramos crear un grupo con un GID en particular nos bastaría el especificar el GID mediante el parámetro *-g*.

sugerencia
Una practica habitual de buena administración es el establecer rangos para los GID de los grupos.Por ejemplo del GID 0 al 100 para grupos administradores, ...

8.2.3. Modificando grupos del sistema

Sólo el usuario *root* puede modificar grupos del sistema. Aunque puede conceder privilegios a otros para hacerlo.

Para esto se utiliza el comando **groupmod**.

sugerencia
man groupmod

8.2.4. Borrando grupos del sistema

Sólo el usuario *root* puede borrar grupos del sistema. Aunque puede conceder privilegios a otros para hacerlo.

Para ello se utiliza el comando **groupdel**.

sugerencia
man groupdel

8.3. Gestión de usuarios

8.3.1. Zona de disco reservada a cada usuario

Cada usuario tiene un espacio en disco para tener sus datos. Este espacio es un directorio con su nombre que se encuentra en `/home/`.

Es posible encontrarlo también de las siguientes formas:

```
/home/b/pcm
/home/b/be/pcm
/home/futurama/pcm
/home/futurama/b/pcm
/home/futurama/b/be/pcm
...
```

Este tipo de estructuraciones se utilizan para una mejor organización de los usuarios y también para evitar exceder el número de entradas por directorio en aquellos sistemas con muchos usuarios.

El directorio personal suele estar almacenado en la variable de entorno *$HOME* y también se le conoce como "~".

importante
Los administradores suelen establecer cuotas de disco en `/home/` para evitar que unos pocos usuarios monopolicen el uso del disco.

8.3.2. El fichero `/etc/passwd`

Este fichero guarda la información relativa a los usuarios del sistema. Debido a problemas de seguridad ahora se utiliza también el fichero `/etc/shadow`.

Una entrada típica:

```
pcm:x:501:501:Bender:/home/pcm:/bin/bash
```

Este fichero contiene una línea por cada usuario del sistema y cada línea son varios campos separados por ":":

1. El primer campo es el nombre del usuario.
2. El segundo campo contenía el hashing de la contraseña del usuario. Es practica habitual el utilizar el *shadowing* de contraseñs y esta información se encuentra ahora en el fichero `/etc/shadow` razón por la cual este campo suele contener una "*x*".
3. El tercer campo contiene el UID del usuario.
4. El cuarto campo contiene el GID del usuario.
5. El quinto campo o campo *GECOS*(General Electric Comprehensive Operating Supervisor) contiene información relativa al usuario como nombre, departamento, ...

   sugerencia
   No conviene poner información sensible en este campo ya que es visible por todo el mundo que tenga acceso al sistema. Además si hay activados servicios como *finger* es fácil obtener esa información sin necesidad de tener cuenta en el sistema.

6. En el sexto campo está el directorio personal del usuario.
7. En el septimo campo se encuentra el comando que se ejecutará cuando haya un inicio de sesión por parte del usuario. Si se prentende que el usuario trabaje en la máquina se pone una *shell*.

sugerencia
Podemos deshabilitar temporalmente los accesos de un usuario al sistema añadiendo el carácter "*" como primer carácter del segundo campo.

8.3.3. Añadiendo usuarios al sistema

Sólo el usuario *root* puede añdir usuarios al sistema. Aunque puede conceder privilegios a otros para hacerlo.

Se pueden utilizar dos comandos para ello:

- **useradd** binario para la creación de usuarios.
- **adduser** es un script en PERL para la creación de usuarios.

La forma típica para crear un usuario es:

```
[root@sal]# useradd -m -d /home/pcm -g pcm -G pcm,users,futurama -c "Bender" -s /bin/bash ↩
    pcm
[root@sal]#
```

- *-m* en caso de no existir el directorio del usuario lo crea.
- *-d /home/pcm* indica cual va a ser el directorio del usuario.
- *-g pcm* indica cual es el grupo principal del usuario.
- *-G pcm,users,futurama* indica los grupos a los que perteneceráa el usuario.
- *-c "Bender"* información del campo *GECOS*.
- *-s /bin/bash* indica la shell que utilizará el usuario.
- *pcm* nombre del usuario.

importante
Cuando creamos un usuario se copian en su directorio todos los ficheros del directorio `/etc/skel/`.

sugerencia
Es practica habitual crear un grupo con el mismo nombre que el usuario y utilizar este grupo como grupo primario del usuario. De esta forma se garantiza que sólo este el en ese grupo y pueda controlar mejor quien accede a sus ficheros.

sugerencia
Al igual que con los grupos se suelen reservar rangos para tipos de usuarios.

sugerencia
Podemos utilizar el flag *-u* para indicar el UID del usuario.

importante
Una vez creado el usuario será necesario establecerle un password utilizando el comando **passwd**:

```
[root@sal]# passwd pcm
Enter new UNIX password: *******
Retype new UNIX password: *******
passwd: contraseña actualizada correctamente
[root@sal]#
```

8.3.4. Eliminando usuarios del sistema

Sólo el usuario *root* puede eliminar usuarios del sistema. Aunque puede conceder privilegios a otros para hacerlo.

Para eliminar usuarios del sistema se utiliza el comando **userdel**:

```
[root@sal]# userdel pcm
[root@sal]#
```

De esta forma eliminamos al usuario *pcm* del sistema sin borrar su directorio personal.

sugerencia
Si quisieramos eliminar también su directorio personal tendrímos que haber utilizado el flag *-r*.

importante
No podremos eliminar un usuario si este tiene abierta una sesión. En caso de necesidad podemos desabilitar sus accesos al sistema mediante el uso del carácter "*" y después matar todos sus procesos.

8.3.5. Modificando una cuenta existente en el sistema

sugerencia
man usermod

8.3.6. El comando id

Este comando se utiliza para obtener información sobre el UID y el GID de los usuarios:

```
[root@sal]# id
id jose
uid=1000(jose) gid=1000(jose) grupos=1000(jose),20(dialout),24(cdrom),25(floppy),29(audio) ↩
    ,44(video),46(plugdev),1001(ftp)
[root@sal]#
```

8.4. Permisos en GNU/Linux

Para evitar que otros usuarios accedan a nuestros ficheros GNU/Linux al igual que otros sistemas UNIX utiliza permisos y cada usuario puede acceder unica y exclusivamente a aquellos ficheros para los cuales tiene concedido acceso.

Dado que en los sistemas UNIX todo son ficheros es importante el conocer bien el mecanismo que otorga privilegios para el acceso a los ficheros.

Los permisos de los ficheros se almacenan utilizando un entero de doce bits y se dividen en ternas:

- La terna más significativa se utiliza para especificar unos permisos especiales que son los *SUID*, *SGID* y el *Sticky Bit*.
- La siguiente terna se utiliza para especificar los permisos del propietario del fichero.
- La siguiente terna se utiliza para especificar los permisos del grupo del propietario del fichero.
- La terna menos significativa se utiliza para especificar los permisos del resto de usuarios, es decir de aquellos usuarios que no estan en el grupo del usuario que posee el fichero.

Podemos utilizar **ls -l** para ver los permisos de un fichero:

```
[pcm@sal]$ ls -l admlinux.xml
-rw-r--r-- 1 jose users 1,8K 2007-04-08 18:38 admlinux.xml
[pcm@sal]$
```

La primera columna nos da los permisos. El primer carácter nos indica el tipo de fichero que es:

- - indica un fichero normal.
- *d* indica un directorio.
- *c* indica un dispositivo carácter (monitor, impresora).
- *s* indica un socket.
- *b* indica un dispositivo de bloques (discos).
- *l* indica un enlace.

Los siguientes tres carácteres indican los permisos que tiene el propietario del fichero.

Los siguientes tres carácteres indican los permisos que tiene el grupo del propietario del fichero.

Los siguientes tres carácteres indican los permisos que tienen el resto de usuarios.

8.4.1. Tipos de permisos

Los permisos típicos que nos podemos encontrar son:

- *Lectura* denotado como "*r*".
  - *Fichero:* Podemos leer el contenido del fichero.
  - *Directorio:* Podemos leer el contenido del directorio (mediante **ls** por ejemplo).
- *Escritura* denotado como "*w*".
  - *Fichero:* Podemos modificar el contenido del fichero.
  - *Directorio:* Podemos modificar el contenido del directorio. Podemos crear y borrar ficheros dentro del directorio.
- *Ejecución* denotado como "*x*".
  - *Fichero:* Podemos ejecutar el fichero.
  - *Directorio:* Podemos entrar al directorio (mediante **cd** por ejemplo).

8.4.2. Cambio de permisos

Sólo el *root* y el propietario del fichero podrán cambiar los permisos de los ficheros.

Para hacerlo se utiliza el comando **chmod**.

---

sugerencia
chmod sólo actua sobre ficheros. Si queremos que actue de forma recursiva sobre todos los directorios tendremos que utilizar el flag *-R*.

8.4.2.1. Cambiar permisos de forma intuitiva

Para cambiar permisos de forma intuitiva utilizaremos "*u*" para hacer referencia a los permisos del usuario, "*g*" para hacer referencia a los permisos del grupo y "*o*" para hacer referencia a los permisos del resto de usuarios.

Además utilizaremos "=" para establecer unos permisos en concreto, "+" para añadir permisos a los ya existentes y "-" para quitar permisos:

```
[jose@sal]$ ls -l evms.xml
-rw-r--r-x 1 jose jose 70992 2007-04-09 00:21 evms.xml
[jose@sal]$ chmod u=rwx,g+w,o-x evms.xml
[jose@sal]$ ls -l evms.xml
-rwxrw-r-- 1 jose jose 70992 2007-04-09 00:21 evms.xml
[jose@sal]$
```

---

sugerencia
Si utilizamos el flag *-v* nos informa del resultado.

8.4.2.2. Cambiar permisos en octal

El método anterior es muy intuitivo, pero engorroso. Es posible utilizar notación octal para establecer los permisos. Al principio cuesta acostumbrarse, pero al poco tiempo es más intuitivo que el método anterior.

Se utiliza notación octal porque con tres digitos en binario se pueden representar ocho números diferentes.

Para poner los permisos en octal se pone a uno el permiso a establecer y a cero el que no se quiere conceder:

Para establecer estos permisos en octal:

```
[root@sal]# chmod 754 admlinux.xml
[root@sal]#
```

	Propietario	Grupo	Resto
Permisos	rwx	rwx	rwx
Binario	111	101	100
Octal	7	5	4

Cuadro 8.1: Permisos en octal

8.4.3. Permisos por defecto

Cuando creamos un fichero se crea con unos permisos por defecto. Estos permisos están especificados por el *umask*:

```
[root@sal]# umask
0022
[root@sal]# umask -S
u=rwx,g=rx,o=rx
[root@sal]#
```

Para determinar la máscara a utilizar se hace al contrario que con los permisos, es decir se pone a uno los permisos que se quieren quitar:

	Propietario	Grupo	Resto
Permisos	rwx	rwx	rwx
Binario	000	010	110
Octal	0	2	6

Cuadro 8.2: Máscara en octal

Para establecer la máscara:

```
[root@sal]# umask 026
[root@sal]# umask -S
0026
[root@sal]#
```

8.5. El comando su

Este comando nos permite ejecutar una shell como otro usuario en la sesión activa. Es decir, nos permite asumir la identidad de otro usuario (si conocemos su password claro):

```
[pcm@sal]$ whoami
pcm
[pcm@sal]$ su -
Password: ********
[root@sal]# whoami
root
[root@sal]# pwd
/root
[root@sal]#
```

Para terminar la sesión bastaráa con presionar *Ctrl + D* (fin de fichero) o tecleando **exit**.

importante
En caso de no indicar ningún usuario con el comando **su** se supone que se está intentando asumir la identidad del *root*.

importante
La diferencia entre utilizar **su - usuario** y **su usuario** es que cuando se utiliza **su -** se hace login de la misma forma que si se logeará en la consola, cargando todos los ficheros de configuración de su perfil.

sugerencia
Es posible ejecutar comandos como si fueramos otro usuario utilizando **su**:

```
[pcm@sal]$ su lila -c ''rm -Rf /home/lila''
Password: ********
[pcm@sal]$
```

8.6. El permiso *SUID*

Hay veces que es necesario que un programa se ejecute con los privilegios de su propietario en lugar de con los privilegios del usuario que lo ejecuta. *SUID* es un acrónomio de *S*et *U*ser *ID*.

Un ejemplo es el comando **passwd**, el cual necesita tener privilegios de *root* ya que tiene que acceder a los ficheros `/etc/passwd` o `/etc/shadow` en modo escritura y sólo el *root* puede hacerlo. Si deseamos que un usuario pueda cambiar su password es necesario que este comando se ejecute con los privilegios de su propietario.

```
[pcm@sal]$ ls -lh /usr/bin/passwd
-rwsr-xr-x 1 root root 28K 2007-02-27 08:53 /usr/bin/passwd
[pcm@sal]$
```

Si observamos la salida anterior veremos algo que nos llama la atención. En lugar de tener una "*x*" en el permiso de ejecución del propietario tenemos una "*s*" lo cual nos indica que este programa se ejecutará con los privilegios de su propietario en lugar de con los del usuario que lo está ejecutando.

aviso
Estos programas constituyen un peligro potencial en un sistema ya que si se hace un mal uso de ellos y el propietario tiene privilegios, *root*, el que lo ejecuta podrá hacer en el sistema todo lo que el propietario del programa tenga permitido.

8.6.1. Activación del permiso *SUID*

Como hemos visto cuando establecemos los permisos en octal utilizamos una terna de números. Para los permisos especiales utilizaremos cuatro números. El primero de ellos hará referencia al permiso especial y los otros tres a los permisos normales.

El *SUID* es el bit más significativo de los tres bits utilzados para los permisos especiales, con lo cual si hacemos:

```
[pcm@sal]$ chmod 4755 miprograma
[pcm@sal]$
```

Estaremos activando el permiso *SUID* y estableciendo los permisos *755* al ejecutable **miprograma**.

importante
Este permiso funciona sólo con binarios y no con scripts (excepto con los de PERL).

8.6.2. El permiso *SUID* y los directorios

Este permiso no tiene efecto en los directorios.

8.7. El permiso *SGID*

Este permiso es igual que el *SUID* sólo que en lugar de ejecutar un fichero con los privilegios del propietario se hará con los privilegios del grupo al que pertenezca el fichero. *SGID* es el acrónimo de *S*et *G*roup *I*D.

```
[pcm@sal]$ ls -lh /usr/bin/wall
-rwxr-sr-x 1 root tty 11K 2007-02-21 18:48 /usr/bin/wall
[pcm@sal]$
```

En este caso vemos que el permiso de ejecución para el grupo está marcado con una "*s*" en lugar de con una "*x*". Esto nos indica que tiene activado el permiso *SGID*.

Podemos ver que el fichero pertenece al usuario *root* y al grupo *tty*. Cuando un usuario ejecute el comando **/user/bin/wall** lo ejecutará con sus privilegios de usuario y con los del grupo *tty*.

8.7.1. Activación del permiso *SGID*

Como hemos visto cuando establecemos los permisos en octal utilizamos una terna de números. Para los permisos especiales utilizaremos cuatro números. El primero de ellos hará referencia al permiso especial y los otros tres a los permisos normales.

El *SGID* es el segundo bit más significativo de los tres bits utilzados para los permisos especiales, con lo cual si hacemos:

```
[pcm@sal]$ chmod 2755 miprograma
[pcm@sal]$
```

Estaremos activando el permiso *SGID* y estableciendo los permisos *755* al ejecutable **miprograma**.

importante
Este permiso funciona sólo con binarios y no con scripts (excepto con los de PERL).

8.7.2. El permiso *SGID* y los directorios

Cuando un directorio tiene activado este permiso todos los ficheros que se creen en el pertenecerán al grupo del propietario sin importar cual sea el grupo del usuario que cree el directorio.

sugerencia
Este permiso es de gran utilidad cuando se trabaja en directorios compartidos.

8.8. El *Sticky Bit*

El *Sticky Bit* es el bit menos significativo de los bits que se utilizan para los permisos especiales. Este permiso también es conocido como *bit pegajoso* o *bit de adhesión*.

Cuando este bit está activado el programa que lo tiene activado se queda en memoria incluso después de terminar su ejecución. Esto hará que se ejecute más rápido a costa de un mayor consumo de memoria.

aviso
Un mal uso de este permiso puede saturar el consumo de memoria del equipo.

```
[pcm@sal]$ ls -lh html
drwxr-xr-t 2 pcm pcm 4,0K 2007-04-10 10:09 html
[pcm@sal]$
```

El permiso de ejecución está con "*t*" en lugar de con "*x*". Esto nos indica que tiene activado el *Sticky Bit*.

8.8.1. Activación del *Sticky Bit*

Como hemos visto cuando establecemos los permisos en octal utilizamos una terna de números. Para los permisos especiales utilizaremos cuatro números. El primero de ellos hará referencia al permiso especial y los otros tres a los permisos normales.

El *Sticky Bit* es el bit menos significativo de los tres bits utilzados para los permisos especiales, con lo cual si hacemos:

```
[pcm@sal]$ chmod 1755 miprograma
[pcm@sal]$
```

Estaremos activando el *Sticky Bit* y estableciendo los permisos *755* al ejecutable **miprograma**.

8.8.2. El *Sticky Bit* y los directorios

El *Sticky Bit* se utiliza sobre los directorios para tener una mayor seguridad sobre los ficheros contenidos en él.

Cuando un directorio tiene activado este permiso no importan los permisos que tengan los ficheros en el contenidos. Sólo el propietario del fichero y el *root* podrán borrar ficheros.

Este permiso permite que todos los usuarios con acceso a un directorio puedan modificar el contenido de los ficheros pero que no puedan borrarlos a menos que sean su propietario.

Capítulo 9

Auditoria y Logs

Es necesario que en los sistemas quede auditado quien está, estuvo y que hizo en el sistema con el mayor detalle posible.

9.1. Usuarios presentes en el sistema

Existen varios comandos para comprobar la presencia de usuarios con sesiones abiertas en el sistema.

9.1.1. El comando who

Nos da información sobre quién está conectado en el sistema:

```
[root@sal]# who
bender :0           2007-04-11 08:06
bender pts/1        2007-04-11 12:50 (:0.0)
[root@sal]#
```

Existen varios flags interesantes:

- *-H* nos muestra las cabeceras de las columnas:

```
[root@sal]# who -H
NOMBRE   LINEA       TIEMPO           COMENTARIO
bender :0           2007-04-11 08:06
bender pts/1        2007-04-11 12:50 (:0.0)
[root@sal]#
```

- *-u* nos muestra el tiempo que estuvo inactivo el terminal.

```
[root@sal]# who -Hu
NOMBRE   LINEA        TIEMPO       INACTIV        PID        PID COMENTARIO
bender pts/0        Apr 11 08:11   .         15772 (192.168.32.60.64)
bender pts/1        Apr 11 08:12 03:44       18528 (192.168.60.64)
zoidberg  pts/3        Apr 11 14:36 01:09       25822 (192.168.60.14)
[root@sal]#
```

- *-q* muestra el número total de usuarios conectados.

```
[root@sal]# who -p
bender bender zoidberg
# users=2
[root@sal]#
```

9.1.2. El comando w

Este comando nos indica lo que está haciendo cada usuario:

```
[root@sal]# w
USER     TTY      FROM              LOGIN@   IDLE   JCPU   PCPU WHAT
root :0       -                08:06   ?xdm?   5:58m  0.01s /bin/sh /usr/bin/x-session- ↩
    manager
root pts/1    :0.0             12:50    1.00s  0.20s  0.00s w
[root@sal]#
```

La información que aparece en las columnas es la siguiente:

- *USER* usuario.
- *TTY* terminal en el que está conectado el usuario.
- *FROM* desde donde está conectado el usuario.
- *LOGIN@* hora en la que empezó la sesión.
- *IDLE* tiempo que el usuario ha permanecido inactivo.
- *JCPU* tiempo total de CPU para todos los procesos en el terminal.
- *PCPU* tiempo total de CPU para todos los procesos activos en el terminal.
- *WHAT* comando que está siendo ejecutado en el terminal.

---

sugerencia
Si le especificamos como parámetro un usuario veremos únicamente la información referente a ese usuario.

9.1.3. El comando users

Este comando nos indica los usuarios que están conectados en el sistema.

```
[root@sal]# users
bender bender zoidberg lila fry fry
[root@sal]#
```

9.1.4. El fichero `/var/run/utmp`

Este fichero contiene los usuarios que están presentes en el sistema en ese momento. Este fichero es utilizado por comandos como **who**, **w**, **users**, **finger** y **write**.

9.2. Usuarios que estuvieron en el sistema

9.2.1. El fichero `/var/log/wtmp`

En este fichero se almacenan las conexiones, mediante login, realizadas con éxito. Es un fichero con formato binario y para leerlo tendremos que utilizar el comando **last**.

importante
Cada vez que se apaga el sistema se logea una entrada con el usuario *reboot*. De esta forma podemos ver los reincios de la máquina.

importante
Si el fichero no se encuentra en el sistema no se logea la actividad.

importante
Este fichero es de acceso en modo lectura para todos los usuarios del sistema.

9.2.2. El commando *last*

Permite ver las conexiones realizadas con éxito a nuestra máquina, si se estan logeando en `/var/log/wtmp`.

```
[root@sal]# last
jose     :0                           Mon Apr  9 19:46   still logged in
reboot   system boot  2.6.17.8        Mon Apr  9 19:46 - 20:42 (2+00:56)
root     tty1                         Mon Apr  9 07:54 - down   (00:04)
reboot   system boot  2.6.17.8        Mon Apr  9 07:51 - 07:58  (00:07)
jose     :0                           Sun Apr  8 16:47 - 00:35  (07:48)
reboot   system boot  2.6.17.8        Sun Apr  8 16:46 - 00:35  (07:48)
jose     :0                           Wed Apr  4 20:30 - 20:57  (00:27)
reboot   system boot  2.6.17.8        Wed Apr  4 20:29 - 20:57  (00:28)
jose     :0                           Fri Mar 30 15:53 - down   (00:17)
reboot   system boot  2.6.17.8        Fri Mar 30 15:52 - 16:10  (00:17)
jose     :0                           Thu Mar 29 20:10 - 02:02  (05:51)
reboot   system boot  2.6.17.8        Thu Mar 29 20:08 - 02:02  (05:54)
rrey     :0                           Thu Mar 29 11:46 - 12:09  (00:22)
reboot   system boot  2.6.17.8        Thu Mar 29 11:46 - 12:09  (00:23)
rrey     :0                           Thu Mar 29 09:47 - 11:37  (01:49)
reboot   system boot  2.6.17.8        Thu Mar 29 09:46 - 11:37  (01:50)
jose     :0                           Wed Mar 28 19:48 - 00:14  (04:26)
reboot   system boot  2.6.17.8        Wed Mar 28 19:47 - 00:15  (04:27)
rrey     :0                           Wed Mar 28 14:30 - 17:02  (02:31)
reboot   system boot  2.6.17.8        Wed Mar 28 14:29 - 17:02  (02:32)
rrey     :0                           Wed Mar 28 09:13 - 09:22  (00:08)
reboot   system boot  2.6.17.8        Wed Mar 28 09:12 - 09:22  (00:09)
rrey     :0                           Wed Mar 28 00:25 - 00:57  (00:32)
jose     :0                           Wed Mar 28 00:09 - 00:24  (00:15)
reboot   system boot  2.6.17.8        Wed Mar 28 00:08 - 00:58  (00:49)
jose     :0                           Mon Mar 26 07:52 - 07:56  (00:03)
reboot   system boot  2.6.17.8        Mon Mar 26 07:51 - 07:56  (00:04)
jose     :0                           Sun Mar 25 23:31 - 00:37  (01:06)
reboot   system boot  2.6.17.8        Sun Mar 25 23:31 - 00:37  (01:06)
jose     :0                           Sun Mar 25 21:28 - 21:55  (00:27)
reboot   system boot  2.6.17.8        Sun Mar 25 20:26 - 21:55  (01:28)
jose     :0                           Thu Mar 22 23:27 - 00:09  (00:42)
reboot   system boot  2.6.17.8        Thu Mar 22 23:24 - 00:09  (00:45)
jose     :0                           Thu Mar 22 07:35 - 07:44  (00:09)
reboot   system boot  2.6.17.8        Thu Mar 22 07:34 - 07:44  (00:10)
```

```
jose     :0                            Tue Mar 20 19:48 - 01:58  (06:09)
reboot   system boot  2.6.17.8         Tue Mar 20 19:47 - 01:58  (06:10)
reboot   system boot  2.6.17.8         Mon Mar 19 22:15 - 01:58 (1+03:43)
jose     :0                            Mon Mar 19 20:12 - down   (01:16)
reboot   system boot  2.6.17.8         Mon Mar 19 20:09 - 21:28  (01:18)
jose     :0                            Mon Mar 19 07:15 - down   (00:40)
reboot   system boot  2.6.17.8         Mon Mar 19 07:15 - 07:56  (00:41)
jose     :0                            Sun Mar 18 22:58 - down   (01:40)
reboot   system boot  2.6.17.8         Sun Mar 18 21:26 - 00:39  (03:13)
jose     :0                            Fri Mar 16 16:04 - down   (02:15)
reboot   system boot  2.6.17.8         Fri Mar 16 16:02 - 18:20  (02:17)
jose     :0                            Fri Mar 16 07:45 - 07:57  (00:12)
reboot   system boot  2.6.17.8         Fri Mar 16 07:44 - 07:58  (00:13)
jose     :0                            Thu Mar 15 22:42 - 22:58  (00:16)
reboot   system boot  2.6.17.8         Thu Mar 15 22:41 - 22:58  (00:17)

wtmp begins Fri Mar  14 07:24:54 2007
[root@sal]#
```

sugerencia
Podemos ver los accesos en función de los terminales *ttyn* pasandole como parámetro a **last** el número de terminal *n*.

sugerencia
Podemos ver las conexiones a los terminales *pts/n* pasandole como parámetro a **last** *pts/n*.

sugerencia
Para aquellas conexiones establecidas en remoto podemos utilizar los flags *-a* y *-d* para conocer el hostname y la IP desde la que se conectaron.

9.2.3. El fichero `/var/log/btmp`

Este fichero es análogo al fichero `/var/log/wtmp` sólo que registra los intentos fallidos de conexión.

9.2.4. El comando lastb

Tiene la misma funcionalidad que el comando **last** sólo que para los intentos fallidos de conexión.

9.2.5. El fichero `/var/log/lastlog`

El fichero almacena la última vez que los usuarios accedieron al sistema. Tiene formato binario con lo cual para consultarlo es necesario utilizar el comando **lastlog**.

9.2.6. El comando lastlog

Imprime por la salida estándar la última vez que un usuario se conecto al sistema.

9.3. Permisos *SUID* y *SGID*

Los ficheros con estos permisos activados es necesario tenerlos controlados ya que se ejecutan con los privilegios de su propietario y no del usuario que los ejecuta. Si el propietario es el *root* un mal uso puede comprometer el sistema.

9.3.1. Peligros con estos permisos

Tengamos el siguiente programa:

```
#include <stdio.h>

#define SIZE 2000

int main (void) {
  FILE *ptFichero;
  char chrBuffer[SIZE];
  int intLeidos;

  /* ABRIMOS EL FICHERO EN SOLO LECTURA */
  ptFichero = fopen("/etc/shadow", "r");

  /* LEEMOS 2000 CARACTERES */
  intLeidos = fread(chrBuffer, sizeof(char), SIZE, ptFichero);

  /* SACAMOS POR LA SALIDA ESTANDAR LOS CARACTERES LEIDOS */
  fwrite(chrBuffer, sizeoff(char), intLeidos, stdout);

  /* CERRAMOS EL FICHERO */
  fclose(ptFichero);

  return 0;
}
```

Y a continuación lo compilamos y hacemos lo siguiente:

```
[root@sal]# cp exploit /media/pendrive
[root@sal]# chmod 4755 /media/pendrive
[root@sal]# ls -lh /media/pendrive/exploit
-rwsr-xr-x 1 root root 1,9K 2007-04-11 12:38 exploit
[root@sal]#
```

Si a continuación llevamos ese pendrive a un equipo en el que tengamos privilegios para montarlo y esté permitido la ejecución de *SUID*. En ese equipo podremos leer el contenido del fichero `/etc/shadow` sin ser el usuario *root*.

Si en lugar de lectura hubieramos programado que sustituyera el password del usuario *root* por uno conocido por nosotros tendríamos acceso a *root* en esa máquina.

aviso
Por cosas como esta es por lo que NUNCA el comando **chown** debería tener activados los permisos SUID.
Aunque las implementaciones de hoy en día de este comando eliminan los permisos *SUID* y *SGID* en implementaciones viejas no lo hacían y era un agujero de seguridad.

9.3.2. Evitando la ejecución de ficheros con esos permisos

La mejor forma es indicar *nosuid* en las opciones en el fichero `/etc/fstab` de los sistemas de ficheros en los que no sea necesario ejecuciones *SUID*.

sugerencia
En los sistemas de ficheros que pueden montar los usuarios es más que recomendable utilizar la opció *nosuid* y puede que también *noexec*.

9.3.3. Localizando estos ficheros

Antés de nada recordemos un par de cosas:

- *SUID* los ficheros con este permiso activado tienen permisos mayores, en octal, que 4000.
- *SGID* los ficheros con este permiso activado tienen permisos mayores, en octal, que 2000.
- *Sticky Bit* los ficheros con este permiso activado tienen permisos mayores, en octal, que 1000.

Para localizar a todos los ficheros con permiso *SUID* activado:

```
[root@sal]# find / -perm +4000 -exec ls -l {} \;
```

9.4. El demonio *syslogd*

Este demonio es utilizado por el resto de demonios para logear sus actividades en los ficheros genéricos de log del sistema.

9.4.1. Las facilidades de *syslogd*

Las facilidades describen quien origina el mensaje y son:

- *auth* mensajes de seguridad y autenticación. En desuso.
- *authpriv* igual que el anterior.
- *cron* mensajes originados por el demonio *crond*.
- *daemon* mensajes originados por otros demonios del sistema.
- *kern* mensajes originados por el núcleo del sistema.
- *lpr* mensajes originados por el demonio de impresión.
- *mail* mensajes originados por el demonio del correo.
- *news* mensajes originaods por el demonio de noticias.
- *security* igual que privauth. En desuso.
- *syslog* mensaje generados por el demonio *syslogd*.
- *user* mensajes genéricos de usuario.
- *uucp* mensajes generados por el demonio *uucpd*.
- *local0,...,local7* reservados para uso del administrador.

9.4.2. Los tipos de *syslogd*

Nos indican los tipos de cada mensaje:

- *none* no envía ningún mensaje.
- *debug* mensajes de depuración.
- *info* mensajes de información.
- *notice* mensajes que necesitan una atención especial.

- *warning* mensajes de aviso.
- *warn* mensajes de aviso. En desuso.
- *err* mensajes de error.
- *error* mensajes de error. En desuso.
- *crit* mensajes críticos, fallo de hardware.
- *alert* mensajes de emergencia. El sistema no está disponible debido a un fallo grave.
- *panic* mensajes de emergencia. En desuso.

9.4.3. El fichero `/etc/syslog.conf`

Este fichero contiene la configuración del demonio *syslogd* y le dice que mensajes tiene que almacenar y donde hacerlo.

En cada línea del fichero se especificará como tratar a los mensajes. Lo más normal seráa indicar una facilidad seguida de un punto y un tipo. Es posible utilziar el asterisco para hacer referencia a todas las facilidades o a todos los tipos.

Veamos algunos ejemplos:

```
*.info;mail.none;authpriv.none;cron.none              /var/log/messages
```

Esto hace que se logee en el fichero `/var/log/messages`:

- Todos los mensajes del tipo info.
- Ningún mensaje del demonio de correo.
- Ningún mensaje de seguridad o autenticación.
- Ningún mensaja del demonio *cron*.

importante
La separación entre las dos columnas se tiene que hacer con tabuladores.

```
*.emerg                         *
```

hace que cualquier mensaje del tipo *emerg* sea notificado con un mensaje de broadcast a todos los usuarios en la red.

```
mail.*;mail.!=info              /var/log/mail
```

Hace que todos los mensajes del demonio de correo, exceptuando los del tipo *info*, se almancenen en el fichero `/var/log/mail`.

9.5. Rotado de logs

Para el rotado de logs se utiliza el demonio *logrotate*.

Logrotate ha sido diseñado para facilitar la administración mediante el rotado de logs.

Logrotate permite el rotado, compresión, borrado y envío de logs. Se definen políticas para cada archivo con su periocidad y caracteríticas.

9.5.1. El fichero `/etc/logrotate.conf`

Este fichero es el fichero de configuración de *logrotate*.

Un archivo típico:

```
# see "man logrotate" for details
# rotate log files weekly
weekly

# keep 4 weeks worth of backlogs
rotate 4

# create new (empty) log files after rotating old ones
create

# uncomment this if you want your log files compressed
#compress

# packages drop log rotation information into this directory
include /etc/logrotate.d

# no packages own wtmp, or btmp -- we'll rotate them here
/var/log/wtmp {
    missingok
    monthly
    create 0664 root utmp
    rotate 1
}

/var/log/btmp {
    missingok
    monthly
    create 0664 root utmp
    rotate 1
}
```

Este archivo consta de opciones globales y luego las opciones por cada archivo a rotar.

Es practica habitual el crear ficheros para los ficheros de log a rotar y almacenarlos en un directorio que tipicamente es `/etc/logrotate.d/`.

Un ejemplo típico para rotar los log de *apache* es:

```
/var/log/apache2/*.log {
        weekly                  # rotado semanal
        missingok               # continuar sin error si el fichero no existe
        rotate 52               # numero maximo de ficheros rotado
        compress                # comprimir cuando se rote
        notifempty              # no rotar el log si esta vacio
        create 640 root adm     # permisos, propietario y grupo del fichero rotado
        sharedscripts           # los scripts de rotado se ejecutan una sola vez
        postrotate              # tarea a realizar despues del rotado
                if [ -f /var/run/apache2.pid ]; then
                        /etc/init.d/apache2 restart > /dev/null
                fi
        endscript
}
```

9.5.2. Ejecución de *logrotate*

Logrotate se ejecuta como tarea bajo *cron*:

```
#!/bin/sh

test -x /usr/sbin/logrotate || exit 0
/usr/sbin/logrotate /etc/logrotate.conf
```

Capítulo 10

Servicios

Denominamos servicios en Linux a las aplicaciones, o conjunto de ellas, que están arrancadas esperando a ser utilizadas, o llevando a cabo tareas esenciales en "background".

Al ser Linux un sistema orientado a redes una parte importante son servicios de red, que utilizan el protocolo tcp/ip. Por ser tema a parte la administración de redes sólo describiremos los servicios sin entrar en al administración.

10.1. Generalidades

Una vez que instalamos un servicio en Linux, normalmente debe ser configurado. Los servicios se deben configurar normalmente con sus parámetros de arranque, y también suele crear un fichero o un directorio en el directorio `/etc`.

Como el servicio será arrancado en el inicio de sistema, se suele crear un script de inicio en el directorio `/etc/init.d`. Es un shell script arrancable que recibe un parámetro que puede ser: start, stop, restart y status. Algunos servicios pueden tener otros parámetros, si lo ejecutamos sin parámetros nos mostrará los posibles.

- Con *start* arrancamos el servicio. Este parámetro es obligatorio en cualquier script de arranque. Dependiendo el servicio que sea configurará algunas características el servicio y ejecutará un proceso que quedará como demonio (proceso en background con unas características especiales).
- *Stop* nos permite parar el servicio. Si existe un proceso en background del servicio, este será retenido. Es muy habitual que el servicio cuando arranca deje un PID en un fichero, que permitirá al script localizar y matar el proceso.
- *Restart* realiza una parada y luego arranca el sistema de nuevo.
- *Status* nos indica si el servicio esta arrancado.

Para arrancar por ejemplo el servicio de base de datos mysql, tendríamos:

```
[sal]# cd /etc/init.d
[sal]# ./mysql
Usage: /etc/init.d/mysql start|stop|restart|reload|force-reload

[sal]# ./mysql start
Starting MySQL database server: mysqld.
Checking for crashed MySQL tables in the background.
[sal]#
```

Siempre que queramos podemos ir al directorio donde se encuentra el script de arranque y ejecutarlo con el parámetro que nos interese. Pero si queremos que el servicio se arranque cuando se inicia el sistema y se pare cuando se apague o reinicie, los script tiene que estar en un directorio preparado para ello. Lo que se hace es enlazar (link) desde el directorio con el servicio en `init.d`. Según el directorio donde lo pongamos se arrancará en un determinado modo de arranque o se parará:

Modo	Directorio	Descripción
1	/etc/rc1.d	Servicios que se arrancan cuando el sistema se arranca en modo de Usuario Único. No se permite conectar nada más que al root desde consola.
2	/etc/rc2.d	Servicios que se arrancan en modo multiusuario, pero sin algunos servicios de red.
3	/etc/rc3.d	Servicios que se arrancan en modo multiusuario, con los servicios de red arrancados, pero en modo consola, sin las X.
4	/etc/rc4.d	No se suele utilizar.
5	/etc/rc5.d	Servicios que se arrancan en modo multiusuario, con los servicios de red y X.
6	/etc/rc6.d	Servicios que se ejecutan en parada o reinicio.

Cuadro 10.1: Tabla de modos

Los modos de arranque de 2 al 4 son configurables por el administrador realmente, lo descrito anterior es la recomendación, pero la configuración por defecto de las distribuciones es diferente en cada una de ellas.

El arranque de los servicios y modo, así como alguna configuración de arranque más lo realiza el proceso init que tiene el fichero de configuración `/etc/inittab`.

Los nombres los links en estos directorios es distinto a como están denominados en `/etc/init.d`, tiene una nomenclatura. La primera letra en una S cuando se llamará al servicio con el parámetro start, y una K cuando se llame con el parámetro stop. A continuación lleva el número en el que se quiere ejecutar el servicio y luego el nombre. Vemos un ejemplo:

```
[pcm@sal]$ cd /etc/rc2.d
[pcm@sal]$ ls -l
lrwxrwxrwx  1 root root 18 Feb 27  2004 S10sysklogd -> ../init.d/sysklogd
lrwxrwxrwx  1 root root 15 Feb 27  2004 S11klogd -> ../init.d/klogd
lrwxrwxrwx  1 root root 13 Feb 27  2004 S14ppp -> ../init.d/ppp
lrwxrwxrwx  1 root root 17 Apr 30  2004 S18portmap -> ../init.d/portmap
lrwxrwxrwx  1 root root 14 Feb 27  2004 S20apmd -> ../init.d/apmd
lrwxrwxrwx  1 root root 26 Apr 24  2005 S20clamav-freshclam -> ../init.d/clamav-freshclam
lrwxrwxrwx  1 root root 16 Feb 27  2004 S20cupsys -> ../init.d/cupsys
lrwxrwxrwx  1 root root 17 Apr 24  2005 S20dirmngr -> ../init.d/dirmngr
lrwxrwxrwx  1 root root 14 Feb 27  2004 S20exim -> ../init.d/exim
lrwxrwxrwx  1 root root 13 Mar  4  2004 S20fam -> ../init.d/fam
lrwxrwxrwx  1 root root 17 Mar  4  2004 S20hddtemp -> ../init.d/hddtemp
lrwxrwxrwx  1 root root 15 Feb 27  2004 S20inetd -> ../init.d/inetd
lrwxrwxrwx  1 root root 19 Feb 27  2004 S20linuxconf -> ../init.d/linuxconf
lrwxrwxrwx  1 root root 14 Mar  5  2004 S20lisa -> ../init.d/lisa
lrwxrwxrwx  1 root root 13 Feb 27  2004 S20lpd -> ../init.d/lpd
lrwxrwxrwx  1 root root 17 Feb 27  2004 S20makedev -> ../init.d/makedev
lrwxrwxrwx  1 root root 15 Apr 24  2005 S20mysql -> ../init.d/mysql
lrwxrwxrwx  1 root root 27 Feb 27  2004 S20nfs-kernel-server -> ../init.d/nfs-kernel-server
lrwxrwxrwx  1 root root 15 Feb 27  2004 S20samba -> ../init.d/samba
lrwxrwxrwx  1 root root 13 Feb 27  2004 S20ssh -> ../init.d/ssh
lrwxrwxrwx  1 root root 18 Apr 30  2004 S20timidity -> ../init.d/timidity
lrwxrwxrwx  1 root root 16 Feb 18 14:06 S20webmin -> ../init.d/webmin
lrwxrwxrwx  1 root root 23 Feb  3  2006 S20wpasupplicant -> ../init.d/wpasupplicant
lrwxrwxrwx  1 root root 20 Mar  4  2004 S21nfs-common -> ../init.d/nfs-common
lrwxrwxrwx  1 root root 18 Oct 24  2004 S21quotarpc -> ../init.d/quotarpc
lrwxrwxrwx  1 root root 17 Feb 29  2004 S21sensord -> ../init.d/sensord
```

```
lrwxrwxrwx  1 root root 15 Apr 24  2005 S50pcscd -> ../init.d/pcscd
lrwxrwxrwx  1 root root 24 Aug 24  2005 S85vpnclient_init -> ../init.d/vpnclient_init
lrwxrwxrwx  1 root root 13 Feb 27  2004 S89atd -> ../init.d/atd
lrwxrwxrwx  1 root root 14 Feb 27  2004 S89cron -> ../init.d/cron
lrwxrwxrwx  1 root root 16 Feb 27  2004 S91apache -> ../init.d/apache
lrwxrwxrwx  1 root root 15 Mar  4  2004 S98local -> ../init.d/local
lrwxrwxrwx  1 root root 13 Feb 27  2004 S99kdm -> ../init.d/kdm
lrwxrwxrwx  1 root root 19 Feb 27  2004 S99rmnologin -> ../init.d/rmnologin
lrwxrwxrwx  1 root root 23 Mar  4  2004 S99stop-bootlogd -> ../init.d/stop-bootlogd
[pcm@sal]$
```

10.2. Servicios de Internet

Internet se basa en protocolos que eran estándares en UNIX, para los cuales UNIX ya contaba con servicios que los implementaban. Esto ha sido una de las principales ventajas del mundo UNIX, de la que Linux ha aprovechado.

10.2.1. apache

Es el servidor para http con más presencia en Internet. Para Linux es el servidor de http por defecto.

Su configuración se realiza normalmente en el directorio `/etc/apache` que contiene el fichero `httpd.conf`.

10.2.2. Correo

En cuanto a los servicios de correo, no se ha llegado a imponer ninguno. Tenemos por ejemplo: exim, postfix, sendmail, etc.

10.2.3. ssh

Para el acceso remoto a Linux en modo consola se ha impuesto el protocolo ssh, que es un telnet encriptado y mejorado. Permite también el envío de ficheros.

La configuración del servicio ssh se realiza en `/etc/ssh` tanto en su modo como servidor, como su configuración por defecto en modo cliente.

10.2.4. xinetd

Es un servicio que nos permite tener disponibles una serie de servicios tcp/udp. Como puede ser finger, telnet, ftp, talk, etc.

Sustituye inetd añadiendo mayor seguridad y control de acceso.

Su configuración se realiza en el fichero `/etc/xinetd.conf`, aunque soporta tener un directorio con cada configuración de los servicios en un fichero.

---

sugerencia
Existe una herramienta para pasar la configuración del viejo inetd a xinetd, llamada **itox**.

10.3. Servicios de Ficheros y Impresión

10.3.1. nfs

Es el sistema de ficheros de red por defecto. En el servidor nos permite mostrar en red un directorio que como cliente podemos montar en nuestro sistema de ficheros.

En el lado servidor sólo tenemos que añadir los directorios a los cuales queremos dar acceso y unos parámetros de configuración en el fichero `/etc/exports`.

En el cliente, bien con la aplicación de montaje de sistemas de ficheros, `mount`, o bien en el fichero de sistemas de ficheros montandos `/etc/fstab` podemos incorporar los directorios exportados por el servidor en nuestra máquina.

10.3.2. samba

El sistema de ficheros de red de Windows se ha impuesto también para su uso como servicio de ficheros. Samba es un proyecto que nos proporciona las herramientas para poder exportar nuestros sistema de ficheros y servicios de red de Windows, y nos permite obtener y montar en nuestro sistema los ficheros compartido.

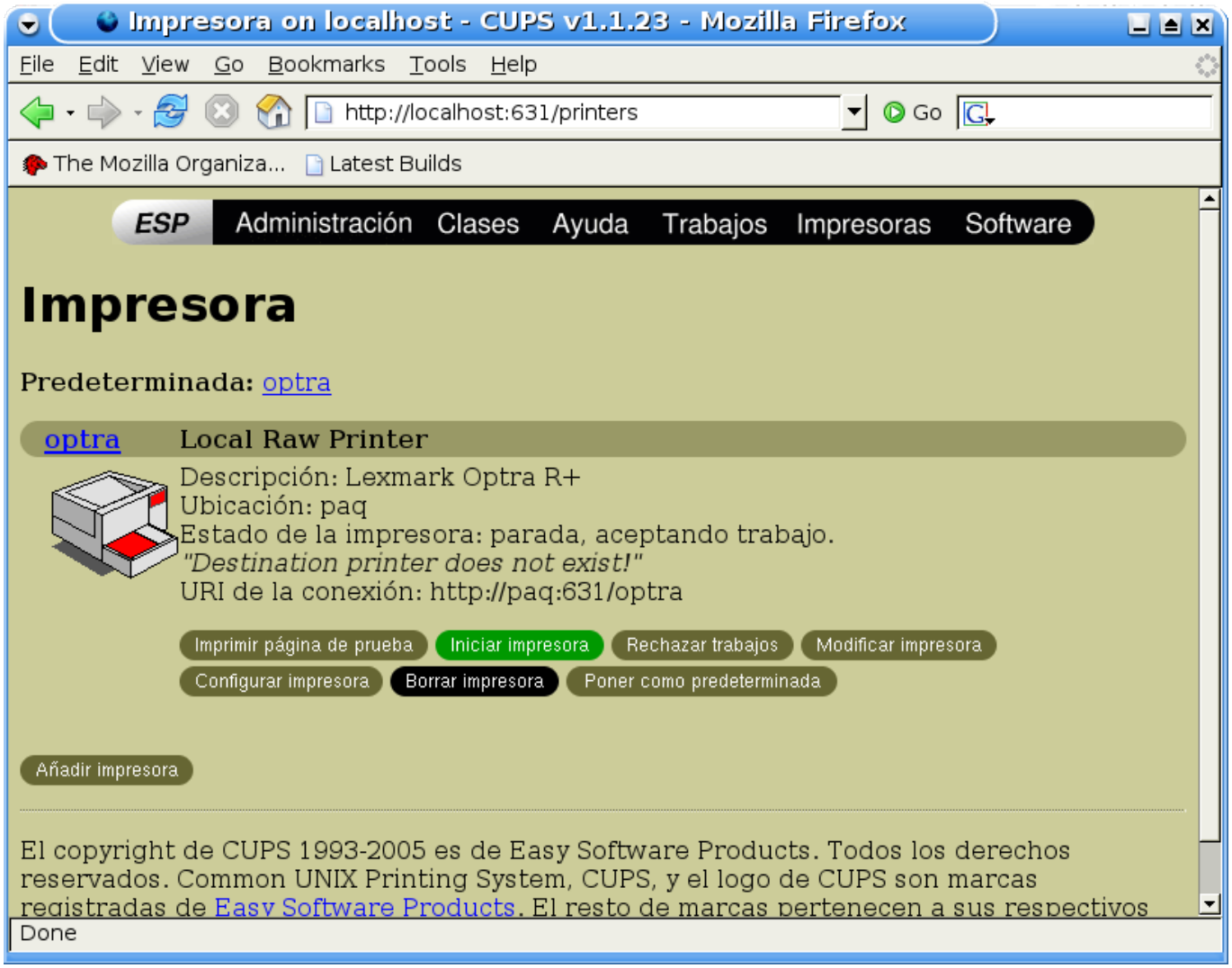
10.3.3. cups

Los servicios de impresión en en las distintas versiones de UNIX fueron los portados en Linux, pero su compleja configuración y realidad con las impresoras que existen actualmente, así como su difícil utilización en red, ha hecho que se esté imponiendo un sistema más reciente y moderno, como es cups (Common UNIX Printing System).

cups es un servicio de impresión que permite configuración remota a través de un servidor web. Utiliza el protocolo Internet Printing Protocol (IPP) para imprimir, aunque soporta la impresión por los comandos tradicionales de UNIX. También dispone de filtros automáticamente y configuración de impresoras a través de Postscript Printer Descriptions (PPD).

La configuración de impresoras se realiza desde la web de administración de cups, pero existe el directorio `/etc/cups` donde se encuentra la configuración del servicio en si.

La pantalla de impresoras de cups tiene está apariencia:



10.4. Servicios de Base de Datos

Como es sabido las bases de datos en una de las aplicaciones más importante de la informática, y los sistemas de gestión de bases de datos (SGBD) disponibles dentro de un sistema operativo es un detalle a analizar, sobre todo si se quiere utilizar Linux como un sistema servidor grande.

Posiblemente Linux se ha incorporado un poco tarde a los servicios de base de datos, pero los resultados son admirables y se esta imponiendo en algunos ámbitos.

Una SGDB importante como Oracle Database ha sido portada a Linux, pero tenemos algunas más, y de código abierto.

10.4.1. mySQL

En las últimas versiones dispone de un motón de características de BD totalmente profesionales.

mySQL es muy utilizado en aplicaciones web, ya que se ha primado mucho la velocidad de acceso al dato, y una menor velocidad en la escritura. Esto es muy común en las aplicaciones web.

Como desarrollo a parte existe un administrador web de mySQL hecho en PHP muy popular, llamado myphpadmin.

10.4.2. PostgreSQL

A partir de un proyecto de la Universidad de Berkeley y debido a que se distribuyó con licencia BSD, se ha continuado pese a que sus desarrolladores iniciales lo había ya abandonado. A lo largo del tiempo se ha creado una SGDB que tiene muchas funcionalidades de las BD profesionales.

Capítulo 11

Interprete de Comandos

Para la relación con el sistema el administrador debe utilizar un interface. Evidentemente en los sistemas modernos los interfaces visuales son la principal forma de dar ordenes al ordenador, pero en Unix y en Linux el administrador debería conocer también un interface textual.

Cuando se utiliza en linea para introducir directamente comandos se denomina shell interactiva.

Para el administrador, el interprete de comandos (shell) y "guiones" del interprete de comandos (shell scripts) son muy importantes por varias razones:

- La mayor parte de herramientas y aplicaciones están preparadas para utilizarse mediante la shell y los scripts.
- La configuración del sistema y de la mayoría de los servicios proporcionados se hacen mediante herramientas proporcionadas en forma de shell scripts.
- La principal forma de automatizar procesos de administración es mediante la creación de shell scripts por parte del administrador.

También veremos algunos servicios donde se utilizan habitualmente los shell script.

11.1. Shell Scripting

Los shell scripts son ficheros de texto que contienen comandos de sistema, comandos propios del interprete de comandos y estructuras de control necesarias para procesar el flujo del programa (tipo while, for, etc). Los ficheros script son directamente ejecutables por el sistema bajo el nombre que se haya dado al fichero. Para ejecutarlos, se invoca el shell junto con el nombre del fichero, o bien se dan permisos de ejecución.

La programación en shell es muy útil y cómoda para crear programas fácilmente modificables, pequeños, no complejos, que resuelvan tareas repetitivas, típicas de los administradores. Además, es un lenguaje preparado para manejar ristras y procesar y filtrar texto, por lo que es mucho más fácil programar en shell, que, por ejemplo, en C.

11.1.1. Algunas shells

Un inconveniente es que no es un lenguaje estandarizado si no que hay varias versiones del shell.

Algunos de los más comunes son:

- El shell Bourne (sh). El shell estándar UNIX, y el que todos los UNIX poseen en alguna versión, en linux es un bash renombrada. El sh fue creado por Stephen Bourne en AT&T a finales de los setenta. El prompt por defecto suele ser un ’$’ y en usario root ’#’.

- El shell Bash (bash). El shell Linux por defecto. Deriva de la bourne shell pero se ha impuesto en gran medida por su utilización en Linux.
- El shell Korn (ksh). Es una mejora del Bourne, escrito en AT&T por David Korn en los años ochenta, intenta combinar la sencillez del Bourne con la eficacia de la shell C, más algún añadido. El prompt por defecto es el $.
- El shell C (csh). Fue desarrollado en la Universidad de Berkeley por Bill Joy a finales de los setenta y tiene unos cuantos añadidos interesantes al Bourne, como un histórico de comandos, alias, aritmética desde la línea de comandos, completa nombres de ficheros y control de trabajos en segundo plano. El prompt por defecto para los usuarios es '%'. Una ventaja de los scripts en C shell es que, como su nombre indica, su sintaxis está basada en el lenguaje C. Como shell posteriores recogen las mejoras de esta, hace que no se utilice mucho, aunque todavía se encuentran muchos scripts desarrollados para esta shell.
- Existen muchas otras que son variantes de estas, normalmente versiones reducidas con aplicaciones específicas.

11.1.2. Creando shell scripts

Cada shell cambia un poco el lenguaje pero tienen muchas características comunes. Vamos a ver un resumen de la sintaxis del lenguaje:

1. Los comentarios se comienzan con #. En la primera linea se debe escribir #! con la shell que (o incluso un interprete, como perl o php) con la que queremos ejecutarla, por ejemplo:

   ```
   #!/bin/bash
   ```

2. Para realizar redirecciones de los programas se utilizan > para salida, < para entrada, 2> para salida de error y |túnel (pipe).

   ```
   cat laza.txt |wc -l > lineas_laza.count
   ```

   La salida del comando **cat** que es le fichero `laza.txt` se le pasa al comando **wc -l** que cuenta las lineas y lo mete en el fichero `lineas_laza.count`.

3. Para definir variables se debe poner el nombre seguido de igual y su valor. Para referenciarlas con el símbolo dolar ($). Existen variables predefinidas, como $1 para el primer parámetro del shell script, $HOME directorio home de usuario, $? código de salida de programa recién ejecutado. Existen muchas más de estas variables dependiendo de la shell.

   ```
   FILE=/tmp/salida
   cat laza.txt | wc -l >> $FILE
   ```

   Crea la variable FILE poniendo un nombre de fichero y la utiliza para añadir la salida del resultado del contador de lineas.

4. Hay tres tipos de comillas, las dobles interpretan las variables que hay dentro, las simples no, y la comilla invertida ejecuta su contenido como un comando y lo mete en la variable.

   ```
   DATE=`date +%d-%m-%Y`;
   MSG1="La fecha es $DATE";
   MSG2='La variable donde guardo la fecha se llama $DATE con el comando
   date +%d-%m-%Y';
   echo $DATE;
   echo $MSG1;
   echo $MSG2;
   ```

   Para ejecutar este script:

   ```
   [pcm@sal]# sh comillas.sh
   14-04-2007
   La fecha es 14-04-2007
   La variable donde guardo la fecha se llama $DATE con el comando date
   +%d-%m-%Y
   [pcm@sal]#
   ```

5. Para las shell la condición verdadera es el 0 y el resto lo interpreta como falso. Existen bastantes operadores para realizar las condiciones. Pueden hacerse condiciones sobre fichero: si es un fichero (-f), si es un directorio (-d), si hay permiso de lectura (-r). También sobre cadenas, sobre números y combinar condiciones.

   Por ejemplo `[ -d .ssh -a \( -n $JDK_HOME -o -n $JAVA_HOME \) ]` nos devolvería como verdadero si existe el directorio `.ssh` y alguna de las dos variables no deben ser vacías.

6. Para el control de flujo tenemos las estructuras `if`, `case`, `while`, `for` y `until`.

7. Existen un conjunto de herramientas que son muy utilizadas en los shell script, como pueden ser `cut`, `grep`, `sed`, `awk`, `date`, etc...

8. Para hacer debug podemos chequear la sintaxis del shell script con:

   ```
   sh -n mishell.sh
   ```

   También podemos hacer que nos muestre la ejecución de los comandos que hay en el shell script y los valores que van tomando las variables con:

   ```
   sh -x mishell.sh
   ```

11.1.3. Ejemplo de un shell script

Como ejemplo de programa shell script vamos a hacer una utilidad para buscar ficheros de texto de DOS en el directorio actual y preguntarnos si lo queremos convertir a fichero de texto UNIX. Los ficheros de texto en la plataformas DOS/Windows para finalizar cada linea llevan dos caracteres de control, el ascii 10 (LF) y el ascii 13 (CR). En cambio en UNIX, y por tanto el Linux los fichero de linea sólo utilizan el carácter de control ascii 10 (LF).

La mayor parte de los editores de Linux ya distinguen si es un texto de DOS o Unix. Además existe un comando para realizar esta conversión, **dos2unix**. Por lo que no suele hacer falta una shell para esta tarea, a no ser que no dispongamos del conversor en el sistema. El programa sería:

```
#!/bin/bash
for fichero in *.txt; do
  if grep ^M $fichero &>/dev/null; then
     resp=x
     while [ $resp != "s" -a $resp != "n" ]; do
       echo "'$fichero' es un fichero texto DOS. convertir? (s/n) "
       read resp
     done
     case $resp in
       s)
         sed 's/^M//' $fichero > /tmp/FILE_TMP
         mv /tmp/FILE_TMP $fichero
         echo "El fichero '$fichero' convertido a texto UNIX";;
       n)
         echo "El fichero '$fichero' se deja texto DOS";;
       *)
         echo "ERROR";;
     esac
   fi
done
```

Primeramente ponemos el comentario para indicar que es un script para bash.

El *for* nos va a realizar un bucle por todos los fichero que terminen en `.txt`.

Hacemos una condición que con el comando **grep** nos mire si el fichero tiene lineas con carácter ascii 13 (CR). Para introducir el carácter ^M hemos pulsado Control+V y Control+M, no se escribe con ^ y la M. Este comando si no encuentra ninguna linea devuelve 1, y si encuentra al menos una linea devuelve cero, con lo cual cumplimos la condición.

A continuación vamos a pedir al usuario que nos confirme la conversión. Para ello ponemos por pantalla la pregunta y con el comando **read** cogemos el valor introducido. Con un *while* insistimos con la pregunta mientras la contestación no sea s o n.

Con *case* comprobamos que ha metido. Sería más lógico hacerlo con un *if else*, pero así vemos está estructura. Sí selecciono *n* se imprime por pantalla que no se hizo nada con el fichero.

Cuando opto por convertir el fichero utilizamos la herramienta **sed** que mediante expresiones regulares nos permite hacer sustituciones dentro de un fichero de texto. En este caso le estamos diciendo que sustituya los CR por nada. La salida la redirigimos a un fichero temporal que luego sustituye al original.

Para ejecutar el programa tendríamos dos posibilidades, o bien lo hacemos ejecutable con el comando **chmod u+w txtunixdir.sh** y luego lo arrancamos como **./txtunixdir.sh** o bien le pasamos a una shell como parámetro nuestro programa:

```
[pcm@sal]# sh txtunixdir.sh
cursos.txt' es un fichero texto DOS. convertir? (s/n)
n
El fichero 'cursos.txt' se deja texto DOS
'lazae11.txt' es un fichero texto DOS. convertir? (s/n)
s
El fichero 'lazae11.txt' convertido a texto UNIX
[pcm@sal]#
```

Si al escribir el programa nos hubiésemos dejado sin poner los dos puntos y coma en las opciones del *case* tendríamos un error, que antes de ejecutar nos lo advertiría.

```
...
  n)
    echo "El fichero '$fichero' se deja texto DOS"
...
```

Si solo queremos comprobar sin ejecutar lo podríamos hacer con la opción -n.

```
[pcm@sal]# sh -n txtunixdir.sh
txtunixdir.sh: line 16: syntax error near unexpected token `)'
txtunixdir.sh: line 16: `       *)'
[pcm@sal]#
```

Corregimos de nuevo, y ahora ejecutamos pero con la opción -x.

```
[pcm@sal]# sh -x txtunixdir.sh
+ grep $'\r' cursos.txt
+ resp=x
+ '[' x '!=' s -a x '!=' n ']'
+ echo ''\''cursos.txt'\'' es un fichero texto DOS. convertir? (s/n) '
'cursos.txt' es un fichero texto DOS. convertir? (s/n)
+ read resp
s
+ '[' s '!=' s -a s '!=' n ']'
+ sed $'s/\r//' cursos.txt
+ mv /tmp/FILE_TMP cursos.txt
+ echo 'El fichero '\''cursos.txt'\'' convertido a texto UNIX'
El fichero 'cursos.txt' convertido a texto UNIX
+ grep $'\r' iptables.txt
+ grep $'\r' lazae11.txt
[pcm@sal]#
```

11.2. Planificación de Tareas

Es muy normal que queramos que nuestros shell script se ejecuten periódicamente o en un determinado momento. Tareas como la realización de backups, borrado de temporales, seguridad se deben planificar mediante estos comandos. Existen unos servicios para temporizar estos trabajos.

11.2.1. at

Mediante el comando at podemos lanzar nuestros procesos a un tiempo determinado sin necesidad de estar conectados en ese momento.

Para realizar una ejecución de un script de backup a las 10 de la mañana haríamos:

```
[pcm@sal]$ at 10am < backup.sh
warning: commands will be executed using /bin/sh
job 11 at 2007-04-01 10:00
[pcm@sal]$ atq
10      2007-04-01 01:00 a pcm
11      2007-04-01 10:00 a pcm
[pcm@sal]$
```

Con el comando `atq` vemos la lista de los procesos pendiente de ejecutarse. Es equivalente utilizar `at -l`.

Si queremos evitar que se ejecute el comando programado podemos borrar la entrada con `atrm` indicando el número de trabajo.

```
[pcm@sal]$ atrm 10

[pcm@sal]$ at -l
11      2007-04-01 10:00 a pcm
[pcm@sal]$
```

El proceso `atd` se encargará de ejecutar nuestra orden en el momento programado.

Existe variante, realmente un script, que permite lanzar nuestra orden solo si la carga del sistema es lo suficientemente baja, configurada en la ejecución de servicio.

Para configurar los usuarios que tienen permisos para utilizar el comando `at` existen dos ficheros `/etc/at.allow` y `/etc/at.deny`. En la ejecución del comando se comprueba:

1. Si existe `/etc/at.allow` y el usuario que ejecuta el comando at o batch está en el fichero se permite ejecución.
2. Si no existe `/etc/at.allow` y existe `/etc/at.deny`, se comprueba que el usuario que ejecuta el comando no este en el fichero, para permitirle.
3. Si no existen ninguno de los dos fichero, solo root puede utilizar el comando.

11.2.2. cron

Este servicio nos permite dejar las tareas programadas. Con el comando at teníamos que estar lanzando el comando por cada ejecución, con cron dejamos ya programado cuando y cada cuanto queremos esa ejecución.

Es uno de los principales recursos de administración ya que como ya hemos comentado las tareas de administración suele ser repetitivas y periódicas, por lo que con este servicio vamos a poder programarlas a nuestro antojo.

El proceso que controla el servicio, es decir, que ejecuta las tareas programadas es `cron`, y para realizar nuestras programaciones debemos utilizar el comando `crontab`.

Para listar tareas utilizaremos `crontab -l`.

```
[pcm@sal]$ crontab -l
*/10 10-19 * * * /root/adsl/vpn.sh >> /root/adsl/vpn.log 2>&1

[pcm@sal]$
```

Vemos que por cada tarea hay una serie de parámetros que debemos entender:

- La primera columna indica el minuto en que se ejecuta, pueden indicarse varios separándolos por comas, por ejemplo 15,30,45, o un rango separando dos valores por un guión. Poniendo el * indicamos que todos los minutos. También se puede indicar cada cuanto, poniendo */2, se ejecutaría cada 2 minutos. Los valores serán entre 0 y 59.

- En la segunda columna indicamos la hora. Igualmente se pueden indicar varias horas separadas por comas, rangos, asterisco y fracciones. Los valores son entre 0 y 23.
- La tercera columna nos indica el día del mes. Los valores pueden ser entre 1 y 31.
- La cuarta nos indica el mes. Los valores son entre 1 y 12, y también se admiten nombres de meses.
- La quita nos indica el día de la semana. Los valores pueden ser entre 0 y 7, siendo el domingo 0 o 7. También admiten los nombres.
- Por último siempre tendremos el comando a ejecutar.

Para crear una nueva tarea utilizaremos o modificar las que tenemos `crontab -e`, que abrirá un editor con las tareas actuales para que modifiquemos o creemos una nueva.

```
[pcm@sal]$ crontab -e
*/10 10-19 * * * /root/adsl/vpn.sh >> /root/adsl/vpn.log 2>&1
~
~
~
~
```

Al igual que el comando `at` la seguridad de utilización del servicio se realiza mediante los ficheros `/etc/cron.allow` y `/etc/cron.deny`. Tenemos que tener en cuenta de nuevo:

1. Si existe `/etc/cron.allow` y el usuario que ejecuta el comando cron está en el fichero se permite la programación.
2. Si no existe `/etc/cron.allow` y existe `/etc/cron.deny`, se comprueba que el usuario que ejecuta el comando no este en el fichero, para permitirle.
3. Si no existen ninguno de los dos fichero, solo root puede utilizar el comando.

11.2.3. anacron

Este comando esta pensado para la automatización de tareas cuando el sistema no esta disponible las 24 horas del día.

Con el servicio cron si en la programación de una tarea el sistema no esta encendido, la tarea no se realizara. Con anacron podemos indicar cada cuanto se realizan ciertas tareas y se encarga de que se lleve acabo con esa frecuencia.

Nos permite configurar las tareas por días, lo normal es que tengamos tareas diarias, semanales (7) o mensuales (30), pero no nos asegura el horario de ejecución, únicamente cuando se inicia el sistema y vea que han pasado esa serie de días ejecutará la tarea.

Capítulo 12

Interfaces de Administración

La gran cantidad de servicios, recursos y comandos que hay que tener en cuenta para administrar actualmente un sistema con Linux, ha hecho que aparezcan distintas aplicaciones que nos permiten tener centralizada la mayor parte de las tareas habituales de administración.

Además las distribuciones, e incluso los entornos gráficos principales, en su intento de simplificar al máximo los procesos de administración han creado y colaborado con la aparición de este tipo de herramientas, Yast para SuSE, Control Center para Mandriva o el Control Center de KDE.

Al ser muy variable las posibles configuraciones de Linux hace que estos interfaces sean orientados a administrar solo algunos aspecto. Así nos encontramos que los interfaces de administración de los entornos gráficos gestionan mayormente aspectos para una configuración de workstation.

Comentaremos las dos más conocidas, genéricas, modulares y amplias.

12.1. webmin

Webmin es una cómoda herramienta ya que nos va a permitir administrar nuestra máquina desde nuestro navegador favorito. Incluso al ser un servidor web podemos realizar la administración remotamente.

Soporta un gran número de distribuciones e incluso otros sistemas operativos distintos a GNU/Linux.

12.1.1. Instalación

En algunas distribuciones como Debian ya viene como un paquete lo que nos va a ahorrar el tener que configurar algunos parámetros.

No esta incluido dentro la distribución SuSE por lo que deberemos bajárnoslo de www.webmin.com.

Una vez descargado lo descomprimimos, entramos en el directorio creado y ejecutamos el comando de instalación.

```
sal:/usr/src # tar -xzvf webmin-1.330.tar.gz

sal:/usr/src # cd webmin-1.330

sal:/usr/src/webmin-1.330 # ./setup.sh
```

Nos ira preguntando la configuración básica de webmin.

```
*********************************************************************
*            Welcome to the Webmin setup script, version 1.330
*
*********************************************************************
```

```
Webmin is a web-based interface that allows Unix-like operating
systems and common Unix services to be easily administered.

Installing Webmin in /usr/src/webmin-1.330 ...

************************************************************************
Webmin uses separate directories for configuration files and log
files.
Unless you want to run multiple versions of Webmin at the same time
you can just accept the defaults.

Config file directory [/etc/webmin]:
```

Donde debe guardar la configuración de la aplicación.

```
Log file directory [/var/webmin]:
```

Directorio donde queremos que deje las trazas.

```
************************************************************************
Webmin is written entirely in Perl. Please enter the full path to the
Perl 5 interpreter on your system.

Full path to perl (default /usr/bin/perl):
```

Webmin utiliza perl por lo que tenemos que tener instalado el interprete. Damos la ruta o si es correcta la propuesta pulsamos Intro.

```
Testing Perl ...
Perl seems to be installed ok

************************************************************************
Operating system name:    SuSE Linux
Operating system version: 10.2

************************************************************************
```

Nos detecta correctamente el sistema operativo, la distribución y la versión instalada.

```
Webmin uses its own password protected web server to provide access
to the administration programs. The setup script needs to know :
 - What port to run the web server on. There must not be another
   web server already using this port.
 - The login name required to access the web server.
 - The password required to access the web server.
 - If the webserver should use SSL (if your system supports it).
 - Whether to start webmin at boot time.

Web server port (default 10000):
```

Puerto en el que va a escuchar el servidor web de webmin. Pulsando Intro aceptamos el que nos propone por defecto. Tener en cuenta que el puerto web por defecto en los navegadores es el 80 que normalmente lo tendremos ocupado por apache

```
      Login name (default admin): root
```

Usuario con el que vamos a acceder, por ejemplo root.

```
      Login password:


      Password again:
```

Nos pide repetida la contraseña con la que vamos a entrar.

```
Use SSL (y/n): y
```

Es recomendable aceptar el uso de SSL. Nos lo debe preguntar siempre que tengamos correctamente instalado los paquete de OpenSSL. Si no lo tenemos instalado nos dará el siguiente error:

```
The Perl SSLeay library is not installed. SSL not
available.
```

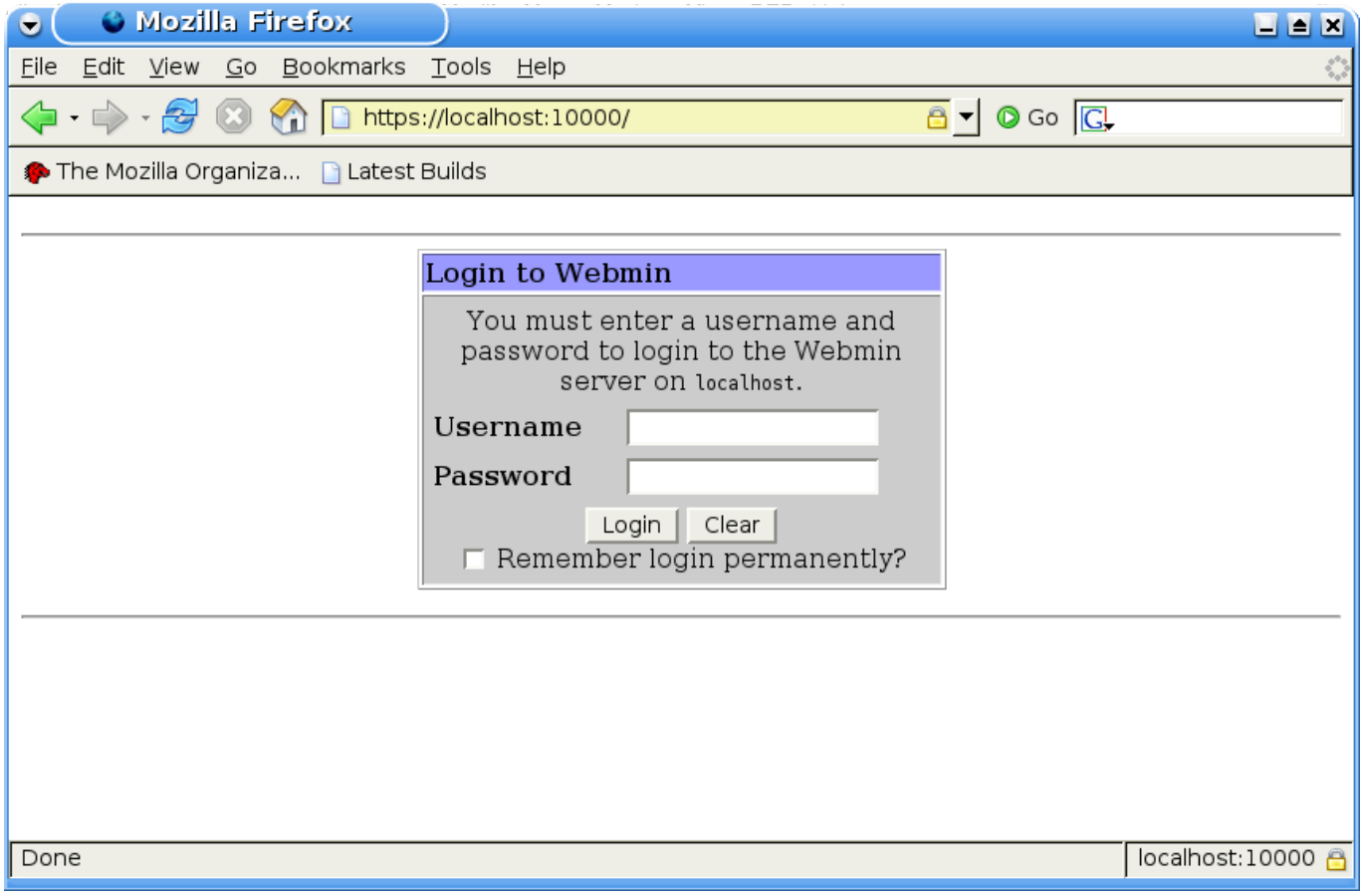
Por último pregunta si queremos que se ejecute como servicio en el arranque del sistema

```
Start Webmin at boot time (y/n): y
```

Para arrancar el servicios debemos ejecutar su script de arranque de `/etc/init.d`, que nos debe haber preparado al decir que queremos arrancarlo en arranque.

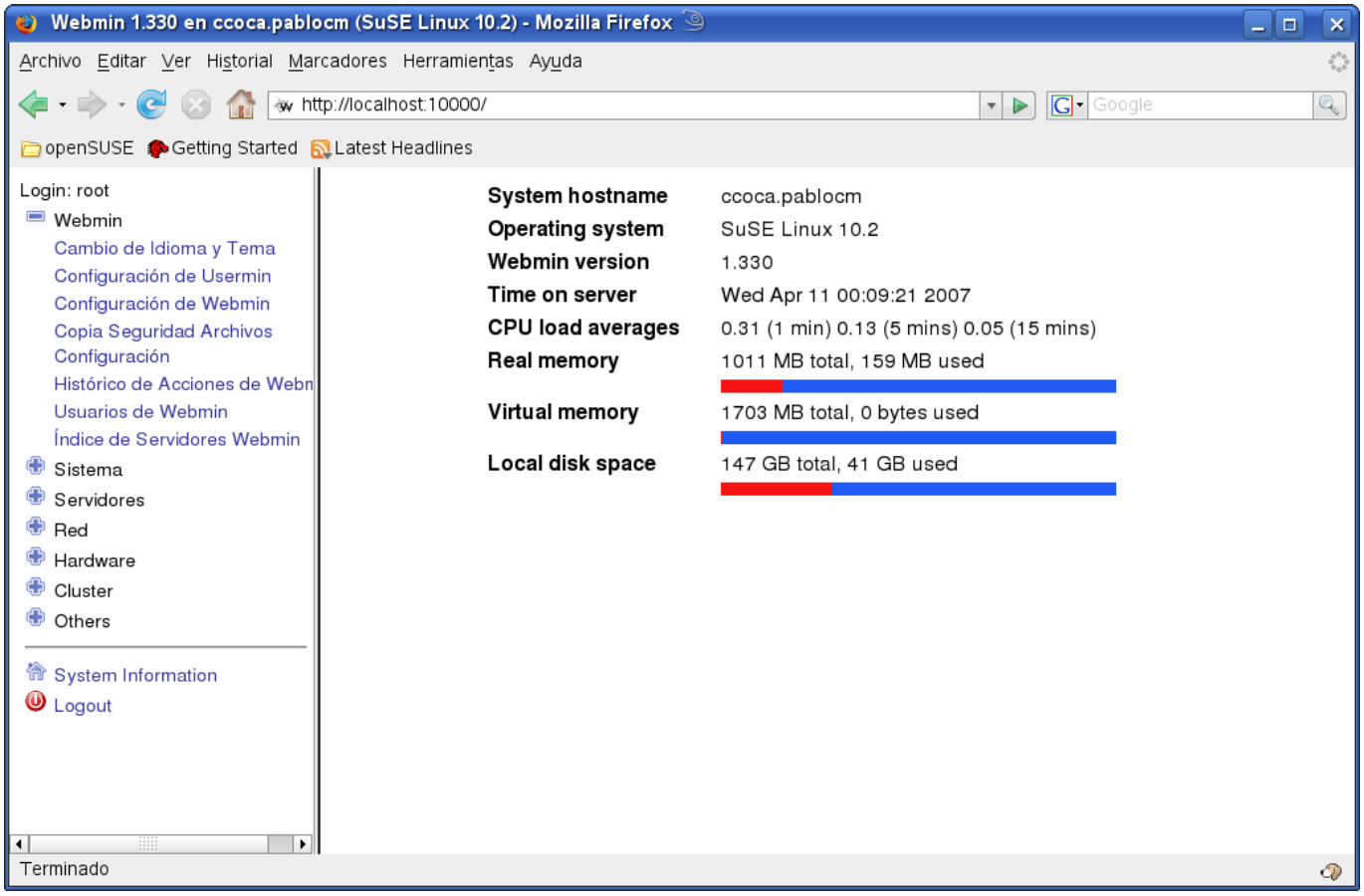
```
sal:/usr/src/webmin-1.330 # /etc/init.d/webmin start
```

Por último vamos a nuestro navegador y tecleamos: `https://localhost:10000` y veremos:



12.1.2. Administración con webmin

Una vez que metemos la usuario que establecimos antes nos aparecerá la pantalla inicial de administración.

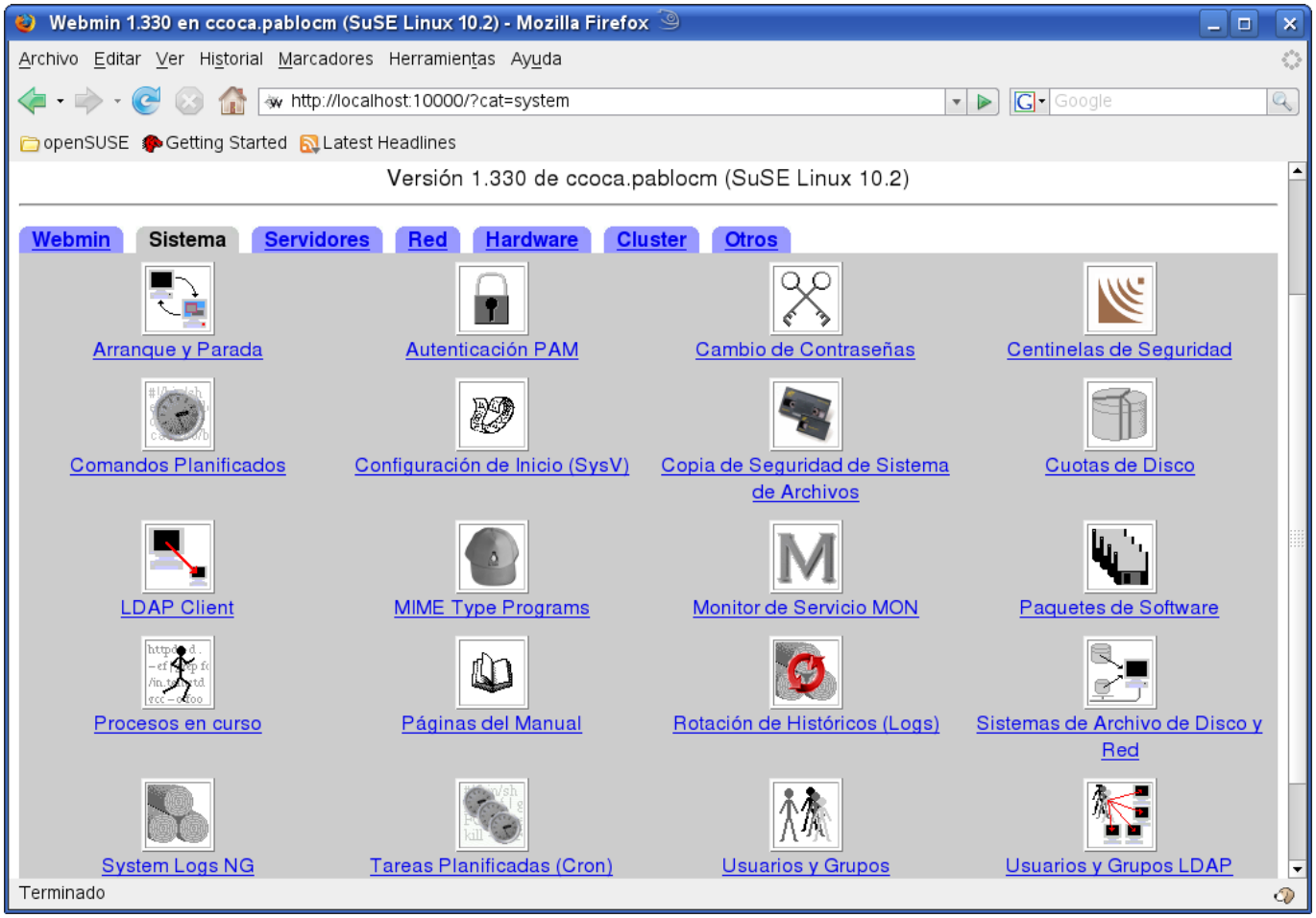


Webmin se presenta con diferentes temas, y también soporta distintos idiomas. En la versión que presentamos por defecto nos muestra un tema en el cual se organiza la aplicación con distintos *frames*, con un menú desplegable a la izquierda y el centro con la selección actual, que inicialmente es información sobre la máquina.

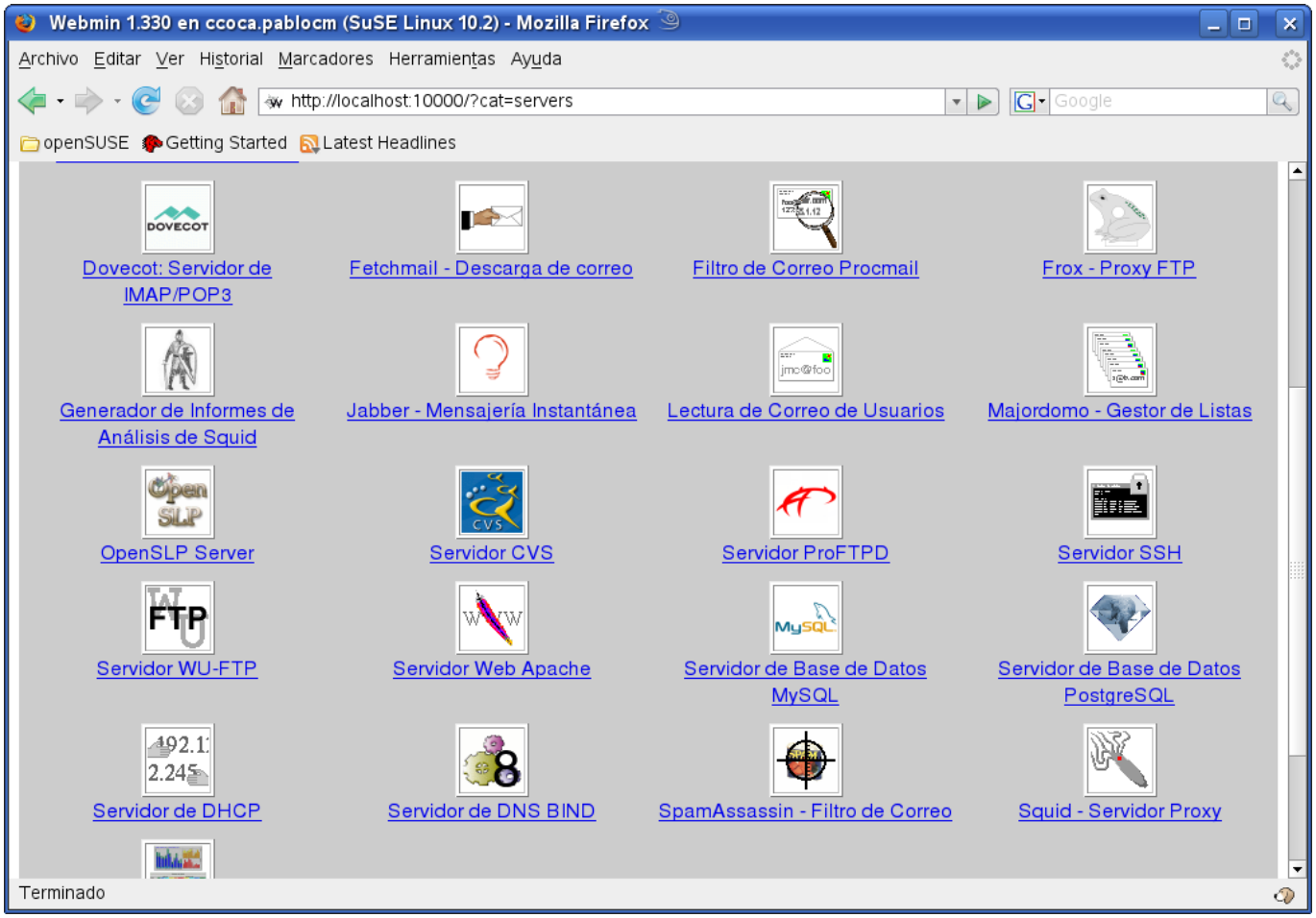
En el menú Webmin tenemos las siguientes opciones:

- *Cambio de Idioma y Tema* nos permite configurar el idioma y la visualización de la aplicación.
- *Configuración Usermin* módulo para configurar una aplicación web para acceso de los usuarios a recursos del sistema con una filosofía parecida a webmin. Permite leer y configurar el correo, configurar los fichero de usuario, etc.
- *Configuración de Webmin* permite la configuración global de la aplicación, IP que pueden acceder, actualización de la aplicación, etc.
- *Copia de seguridad de Archivos de configuración* es un sistema para hacer copias de seguridad de la configuración del sistema.
- *Histórico de Acciones* lleva un control de las acciones que se han hecho con la aplicación.
- *Usuarios de Webmin* usuarios que pueden entrar en Webmin y se puede configurar las opciones (módulos) a las que puede acceder.
- *Índice de Servidores* nos permite acceder a otros servidores webmin de otros sistemas.

La segunda opción es la de Sistema.

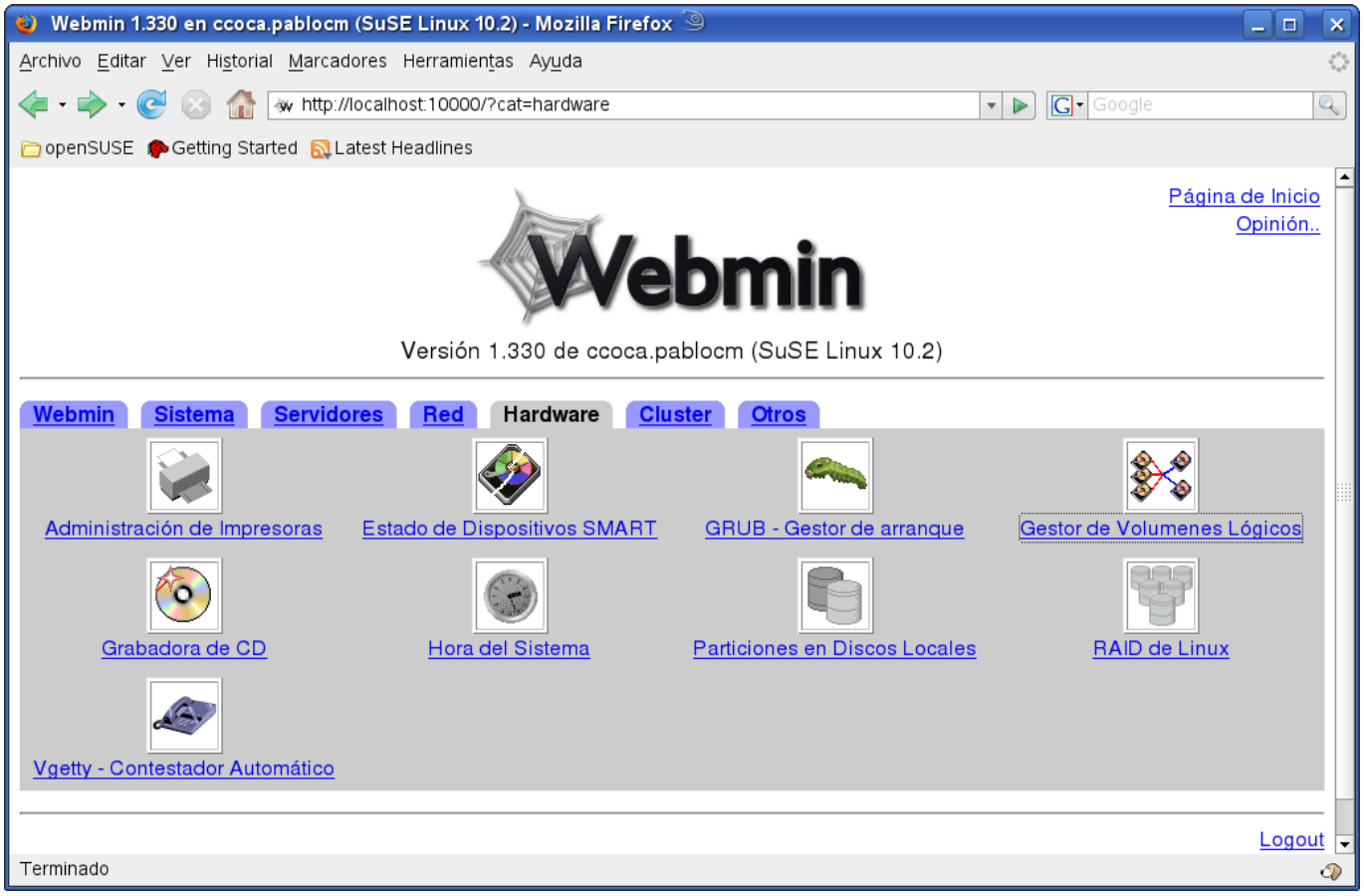


Nos permite el control de todas las tareas que debe hacer el administrador sobre un sistema. Desde este panel controlamos desde copias de seguridad de nuestra máquina a configuración de usuarios de la máquina. Esta vez vemos el tema clásico de webmin.



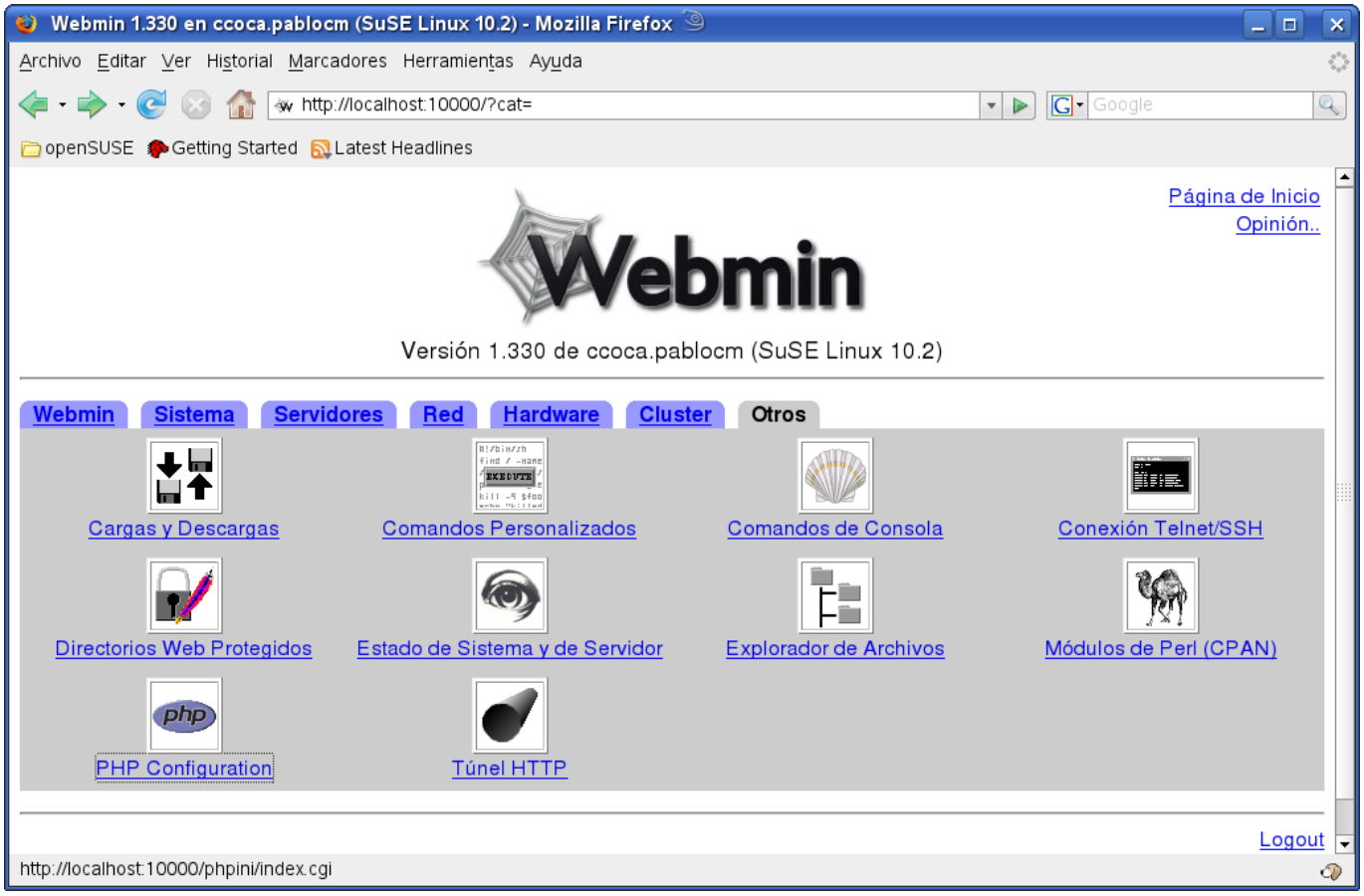
A continuación tenemos administración Servidores, referido a los servicios de la máquina y Red que realmente son servicios, pero están más orientados a niveles más bajos de red, como vpn, contrafuegos, ppp, etc. Entre los servidores se encuentra la configuración de apache, correo, ftp, jabber, bases de datos, etc.

Hay que tener en cuenta que webmin, en su menú de Configuración de Webmin nos permite reordenar los módulos según nuestros criterios.



La opción *Hardware* nos muestra módulos de gestión de distintos componentes físicos de nuestro sistema. Como son los discos, el reloj de sistema, grabadora, etc. Incluye en este apartado las impresoras, permitiéndonos controlar un servicio de impresoras como cups de una manera transparente.

En la pestaña de *Cluster* nos permite administrar distintas máquinas que tengan webmin, realizando los mismos cambios sobre todos las instancias. Por ejemplo podemos ejecutar un script sobre varios o todos de estos sistemas.



Por último en *Otros* tenemos algunas herramientas de administración, como copiar ficheros, lanzar comandos sobre la máquina, conectarnos por ssh con un *applet* java a nuestro sistema, etc.

En resumen, tenemos en nuestro navegador favorito todo aquello que un administrador puede necesitar.

12.1.3. Nuevos módulos

Una de las ventajas de webmin es la posibilidad de añadir nuevos módulos lo que permite administrar servicios que son menos comunes, incluso que el creador del servicio decide implementar el módulo para que se pueda realizar la administración desde webmin

Así podemos ver en la web de webmin los distintos módulos realizados por terceras partes, agrupados por distintas categorías.

12.2. linuxconf

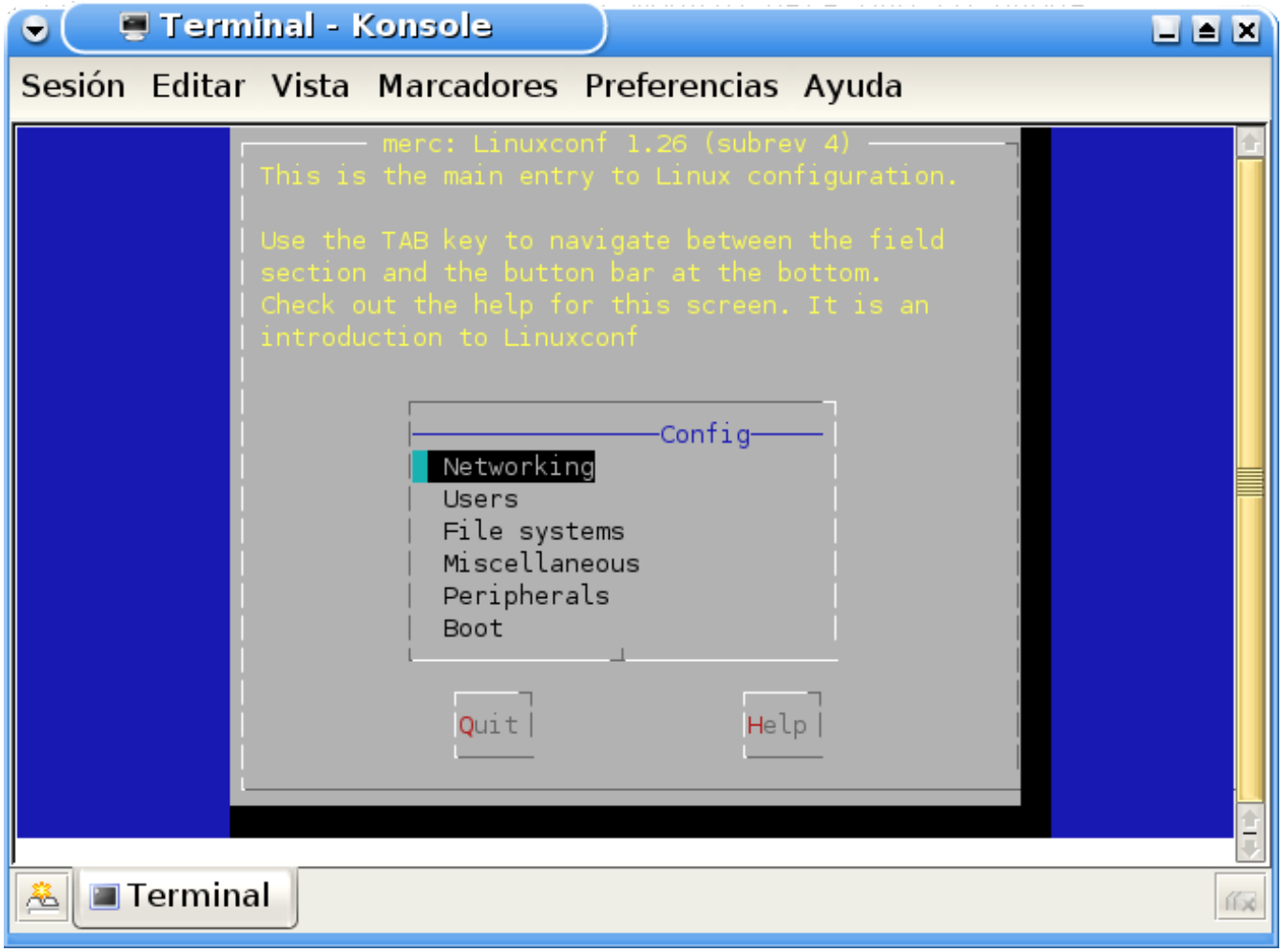
Se trata de otro interface de administración, también modular y aunque su modo de presentación inicial fue en consola textual tiene implementado un interface web.

Resulta más sencillo y es común que este en las distribuciones ya que tiene menos requisitos que webmin.

Para ejecutarlo con el usuario root escribimos en consola:

```
sal:/ # linuxconf
```

Nos aparecerá una pantalla con la siguiente:

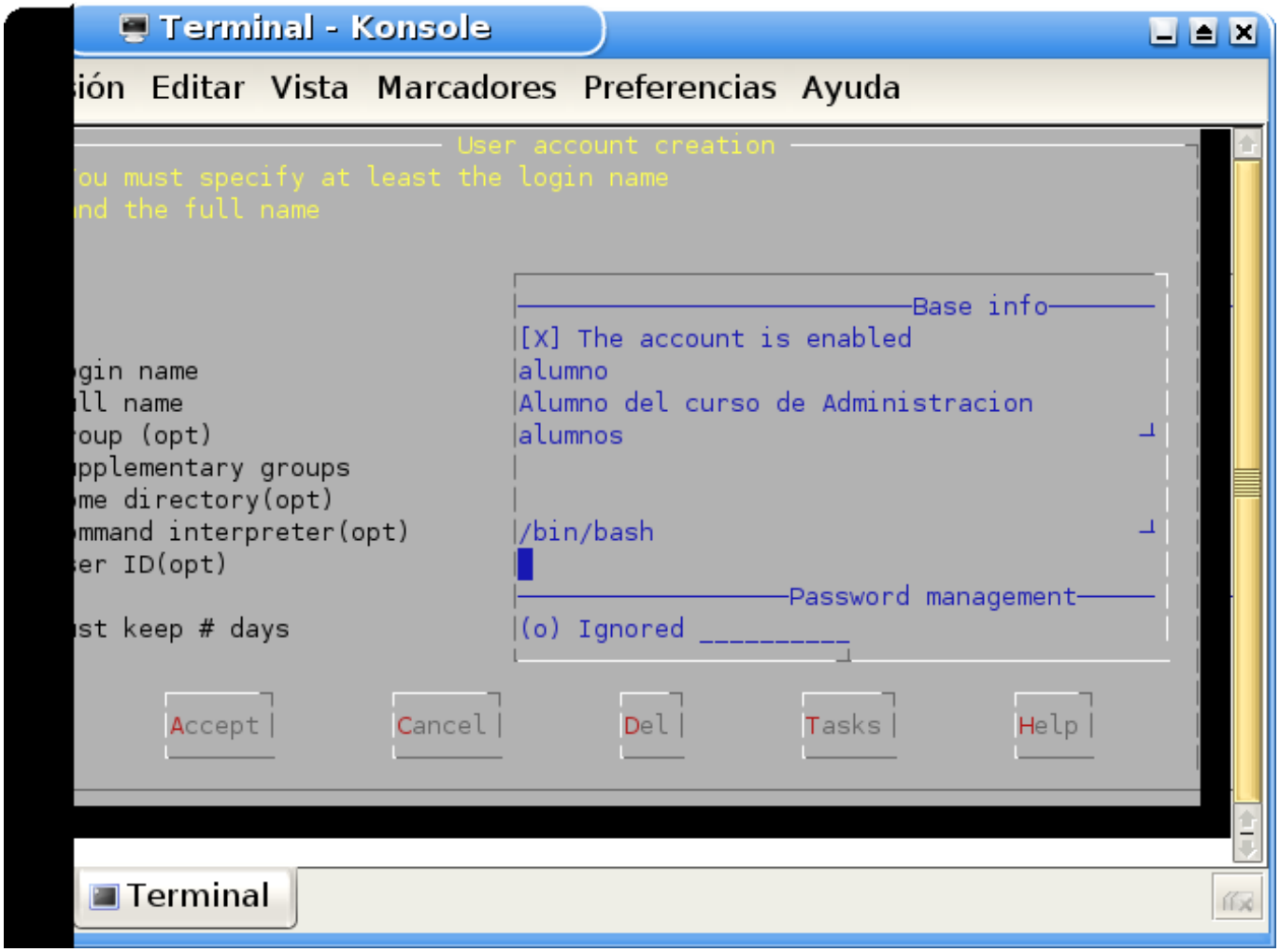


12.2.1. Administración con linuxconf

A partir de la pantalla de inicio podemos navegar con los cursores por las distintas opciones de administración de linuxconf.

Para mostrar la utilización de la aplicación vamos a añadir un usuario, para ello desde la pantalla inicial con la flecha hacia bajo nos desplazamos hasta `Users`, pulsamos Intro y nuevamente sobre `User acounts`, nos aparecerá la lista de usuarios. Con ello lo que ha hecho nuestro interface es mostrarnos en contenido del fichero de administración de usuarios `/etc/passwd`.

Con el tabulador seleccionamos el botón `Add` e Intro. Nos aparece una pantalla para introducir los datos de usuario, rellenando cada campo y moviéndose por ellos con las teclas de cursor. Con el tabulador podemos ir al botón de Accept. La herramienta hará un useradd con los parámetros que hemos rellenado.



Si el grupo no existe, nos pregunta si queremos crearlo y nos pide las claves de usuario, actualizando el fichero de administración de grupos `/etc/groups`. Nos aparecerá la lista de usuarios con el nuevo. Con el botón de Dismiss podemos ir al menú inicial.

Podemos administrar los siguientes aspectos con la configuración los módulos básica de linuxconf:

- La hora de sistema, la zona horaria y el reloj del ordenador.
- Gestor de arranque LILO.
- Configuración de la red básica (número IP, mascara de red, ...).
- Red IPX.
- Rutas de red estáticas.
- Sistemas de ficheros (/etc/fstab).
- Servicio de Enrutado.
- Cliente NIS (ypbind).
- Servidor de ficheros NFS.
- Cliente PPP.
- Cuentas de usuario y grupos.
- Política Shadow.

- Servicio DNS.
- Servicio de correo Sendmail.
- Cortafuegos (Filtrado de paquetes).
- Servicio RARP.
- Servicio DHCP.
- Alias IP.
- Conexión con UUCP.
- Cuotas de Disco.

Linuxconf es capaz de editar algunos ficheros de configuración estándar del sistema, como por ejemplo.

- /etc/fstab
- /etc/host.conf
- /etc/hosts
- /etc/networks
- /etc/resolv.conf

Capítulo 13

Gestión de paquetes

La distribución de software en Linux se realiza mediante paquetes.

Se denomina gestor de paquetes a la colección de herramientas que sirven para automatizar el proceso de instalación, actualización, configuración y eliminación de paquetes de aplicaciones. Como administradores debemos conocer los principales formatos de paquetes y la utilización de las herramientas que los gestionan.

Estas herramientas nos permiten mostrarnos todo el software disponible acudiendo a los repositorios de aplicaciones, ver si tenemos la ultima versión del software y garantizarnos que la dependencia que hay entre distintas aplicaciones por sus versiones es correcta.

La distribución básicamente será la que nos determine, desde que la instalamos por primera vez cual van ha ser las herramientas de gestión de paquetes y obtendrá el software necesario de un repositorio, vía Internet, cd, dvd, etc.

En Linux hay principalmente dos formatos o tipos de paquetes sobre los cuales se desarrollan las herramientas que luego las distribuciones utilizan para instalar el software. Es decir, existen distintas aplicaciones, repositorios, interfaces y variantes para manejar los paquetes pero todos ellos se basan en unos pocos formatos de paquetes.

13.1. rpm

Fue creado por Red Hat y es uno de los más usados por las distribuciones. Es es formato recomendado por Linux Standard Base.

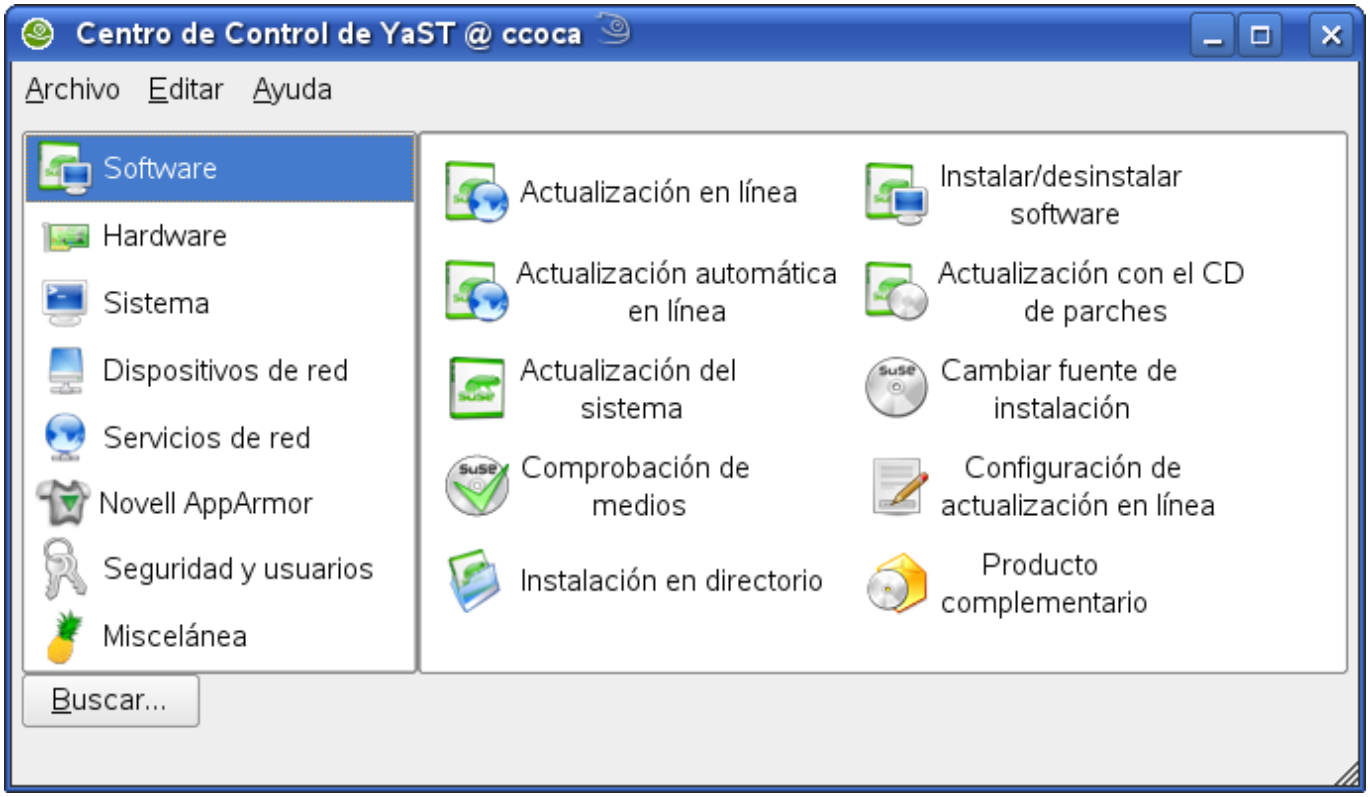
Existen muchas herramientas que los manejan, que se han adaptado sobre todo a cada una de las distribuciones que las utilizan, incluso los gestores gráficos KDE y Gnome disponen de herramientas para su gestión. Pero su utilización en linea es bastante sencilla.

```
rpm -opciones nombre_paquete-version-sistema.rpm
```

Con las opciones controlamos si lo que queremos es instalar, desinstalar, actualizar o información de un paquete. Cuando queremos desinstalar o información de un paquete instalado solo con poner el nombre es suficiente. La forma de nombre, versión, el sistema y la extensión de rpm es la nomenclatura que se utiliza para nombrar los ficheros.

Al contener los paquetes aplicaciones ya compiladas en la etiqueta de sistema se informa para cual están preparados, por lo que nos encontramos .i386 para procesadores 80386 o superiores, i686 para PentiumIII o superior, .ppc para PowerPC, .noarch para indicar que son independientes de la arquitectura (por ejemplo ficheros de texto, script, etc) o .src para rpm que contienen los fuentes de la aplicación. Rpm es capaz a partir de un paquete de fuentes generar el rpm para un sistema en concreto.

La aplicación gráfica de administración de SuSE incluye la gestión de paquetes, soportando el formato rpm.



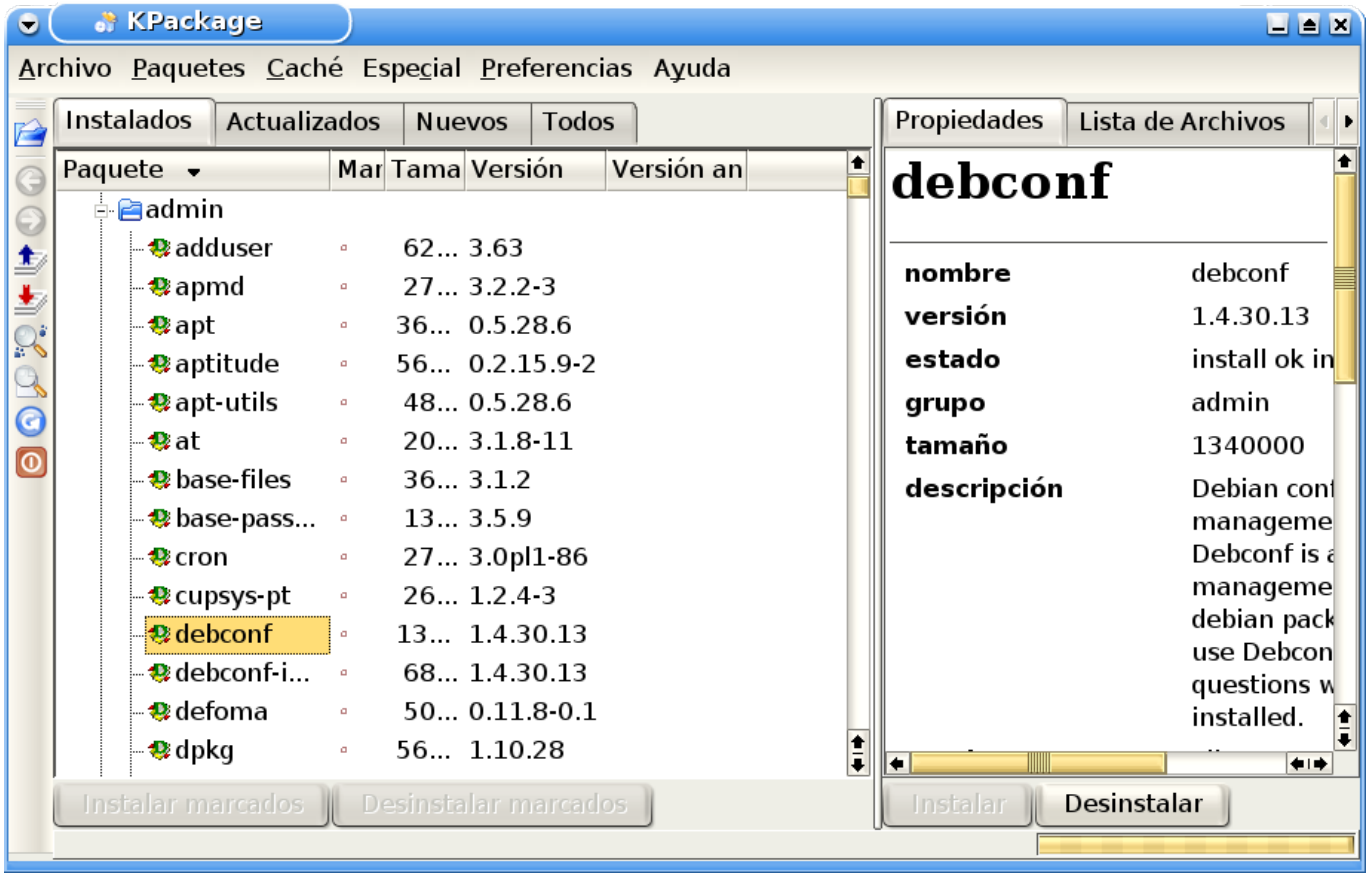
13.2. deb

Es el formato de paquetes creado para Debian. Todas aquellas distribuciones que se basan en Debian, como son Ubuntu o Linex, utilizan este formato de paquetes.

El programa de la distribución Debian que maneja el formato de los paquetes es dpkg. La gestión de los paquetes se realiza con otra aplicación, apt (Advanced Packaging tool), que se encarga de la localización del repositorio de paquetes, ya sea un cd, por una conexión ftp o http. Ambas son un conjunto de herramientas que se utilizan desde consola. Su utilización es algo más compleja que rpm debido a que son varios comandos que cada uno hace una operación distinta.

Existen numerosas interfaces tanto gráficas como textuales que nos permite instalar y gestionar las aplicaciones de una manera sencilla, y se encargan de utilizar el conjunto de comandos de dpkg/apt.

La herramienta de gestión de paquetes del sistema KDE permite la gestión e instalación de este tipo de paquetes.



La estructura del paquete es un fichero ar que contiene, a su vez tres ficheros. Uno con la versión del paquete, otro con la información del paquete y otro con los ficheros que se instalan.

13.3. Otros sistemas

En las primeras distribuciones los paquetes los distribuían en un "tar" y comprimidos. Este sistema no permite tener un control de dependencias, ni gestión sencilla de paquetes instalados, actualizado y su desinstalación. Por lo cual es un sistema demasiado problemático o redundante cuando el sistema tiene gran cantidad de aplicaciones instaladas.

Algunas distribuciones las aplicaciones las distribuyen en paquetes que contienen el código fuente, que al instalarse debe antes compilarse la aplicación, como por ejemplo en Gentoo. La principal ventaja de esta forma de distribuir el software es que la compilación se realiza lo más óptima posible al hardware de la máquina.

Existen también formatos de distribución para linux en sistemas embebidos y PDAs, como es ipk. Este es un derivado simplificado del formato de Debian.

Cuando tenemos una aplicación en un tipo de paquete y estamos interesados en llevarlo a una distribución que utiliza otro sistema, podemos todavía instalarla mediante una conversión con la aplicación alien.

Seguro que alguna vez, si queremos tener la última versión o adaptar la aplicación a nuestras necesidades, y si la aplicación es de código libre, no nos quedará más remedio que coger el código fuente, ya sea bajándonoslo en `tar.gz`, `zip` o del repositorio de cvs o svn.

Apéndice A

GNU Free Documentation License

0. PREAMBLE

The purpose of this License is to make a manual, textbook, or other functional and useful document "free" in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondarily, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of "copyleft", which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

1. APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that work under the conditions stated herein. The "Document", below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as "you". You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

A "Modified Version" of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A "Secondary Section" is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document’s overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The "Invariant Sections" are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The "Cover Texts" are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A "Transparent" copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not "Transparent" is called "Opaque".

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML, PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF and JPG. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word processors for output purposes only.

The "Title Page" means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, "Title Page" means the text near the most prominent appearance of the work's title, preceding the beginning of the body of the text.

A section "Entitled XYZ" means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as "Acknowledgements", "Dedications", "Endorsements", or "History".) To "Preserve the Title" of such a section when you modify the Document means that it remains a section "Entitled XYZ" according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties: any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

2. VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

3. COPYING IN QUANTITY

If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent

copy of the Document, free of added material. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

4. MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.

B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.

C. State on the Title page the name of the publisher of the Modified Version, as the publisher.

D. Preserve all the copyright notices of the Document.

E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.

F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.

G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document’s license notice.

H. Include an unaltered copy of this License.

I. Preserve the section Entitled "History", Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.

J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the "History" section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.

K. For any section Entitled "Acknowledgements" or "Dedications", Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.

L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.

M. Delete any section Entitled "Endorsements". Such a section may not be included in the Modified Version.

N. Do not retitle any existing section to be Entitled "Endorsements" or to conflict in title with any Invariant Section.

Ñ. Preserve any Warranty Disclaimers.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version's license notice. These titles must be distinct from any other section titles.

You may add a section Entitled "Endorsements", provided it contains nothing but endorsements of your Modified Version by various parties--for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

5. COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections Entitled "History" in the various original documents, forming one section Entitled "History"; likewise combine any sections Entitled "Acknowledgements", and any sections Entitled "Dedications". You must delete all sections Entitled "Endorsements".

6. COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

7. AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an "aggregate" if the copyright resulting from the compilation is not used to limit the legal rights of the compilation's users beyond what the individual works permit. When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document's Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate.

8. TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

If a section in the Document is Entitled "Acknowledgements", "Dedications", or "History", the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

9. TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided for under this License. Any other attempt to copy, modify, sublicense or distribute the Document is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

10. FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See http://www.gnu.org/copyleft/.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License "or any later version" applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation.

ADDENDUM: How to use this License for your documents

To use this License in a document you have written, include a copy of the License in the document and put the following copyright and license notices just after the title page:

> Copyright (C) YEAR YOUR NAME.
>
> Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.2 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled "GNU Free Documentation License".

If you have Invariant Sections, Front-Cover Texts and Back-Cover Texts, replace the "with...Texts." line with this:

> with the Invariant Sections being LIST THEIR TITLES, with the Front-Cover Texts being LIST, and with the Back-Cover Texts being LIST.

If you have Invariant Sections without Cover Texts, or some other combination of the three, merge those two alternatives to suit the situation.

If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public License, to permit their use in free software.