6. Gestión de Usuarios

6.1. Creación de Usuarios

Creación Manual de Usuarios

El proceso de creación de usuarios siempre ha sido la tarea más tediosa para un administrador, ya que los pasos a realizar eran largos aunque no complicados, básicamente son:

1. Editar el fichero `/etc/passwd`.
2. Poner un *password* inicial.
3. Crear un directorio propio al usuario con los permisos adecuados.

Además es conveniente hacer:

1. Copiar los ficheros de configuración al usuario.
2. Poner una cuenta de correo y los directorios asociados.
3. Registrar el grupo del usuario en `/etc/group`.
4. Servicios de "*accounting*".
5. Recopilar información de contacto del usuario.
6. Activar cuotas de disco.
7. Verificar que la cuenta funciona.

Por eso normalmente todos los administradores creaban macros para hacer esta trabajo de una forma más o menos automática. Actualmente, casi todos los sistemas disponen de algún comando que realice esta tarea, usualmente "`useradd`" o "`adduser`", e incluso en Red Hat herramientas gráficas como **`Linuxconf`**. Si no disponemos de ellos tendremos que hacer esta labor manualmente.

El primer paso es editar el fichero `/etc/passwd` (en sistemas BSD se puede utilizar el comandode edición ***vipw***). Este fichero contiene, en cada línea, información sobre cada usuario, ya sea real o generado por la instalación del sistema. En cada línea están el nombre de usuario, el *password* encriptado, el UID (número de usuario), el GID (número de grupo de usuario), información de contacto, directorio personal y tipo de *shell* por defecto. A continuación puede verse su aspecto (cuando tenemos *shadow password*):

```
root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:
daemon:x:2:2:daemon:/sbin:
adm:x:3:4:adm:/var/adm:
lp:x:4:7:lp:/var/spool/lpd:
sync:x:5:0:sync:/sbin:/bin/sync
shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown
halt:x:7:0:halt:/sbin:/sbin/halt
mail:x:8:12:mail:/var/spool/mail:
news:x:9:13:news:/var/spool/news:
uucp:x:10:14:uucp:/var/spool/uucp:
operator:x:11:0:operator:/root:
games:x:12:100:games:/usr/games:
gopher:x:13:30:gopher:/usr/lib/gopher-data:
ftp:x:14:50:FTP User:/var/ftp:
nobody:x:99:99:Nobody:/:
xfs:x:43:43:X Font Server:/etc/X11/fs:/bin/false
gdm:x:42:42::/home/gdm:/bin/bash
rpcuser:x:29:29:RPC Service User:/var/lib/nfs:/bin/false
rpc:x:32:32:Portmapper RPC user:/:/bin/false
mailnull:x:47:47::/var/spool/mqueue:/dev/null
rafa:x:500:500:Rafael Menendez de Llano Rozas:/home/rafa:/bin/bash
alumno:x:501:501::/home/alumno:/bin/bash
```

```
jagm:x:502:502::/home/jagm:/bin/bash
```

El nombre de usuario tendrá hasta 8 caracteres (los demás se ignorarán), pudiendo contener (pero no empezar) números, pero no signos de puntuación. Mayúsculas y minúsculas son diferentes (hay que tener en cuenta que normalmente el nombre de usuario será el mismo que la cuenta de correo). Un método que se suele usar es utilizar el primer apellido antecedido de la inicial del nombre (por supuesto hay todas las variaciones que quiera el administrador), si esta combinación resulta larga (más de 8 caracteres) se pueden hacer una entrada en aliases, por ejemplo Rafael Menéndez tendría un login rmenendez de 9 caracteres, entonces podemos crear una cuenta rafam y escribir en el fichero de alias: `/etc/aliases` una línea como:

```
rmenendez:   rafam
```

A partir de ahora tendremos dos nombres para una sola cuenta, al menos en lo que al correo se refiere, por lo que tendremos que tener la precaución de no usar ninguna de ellas de nuevo en una nueva cuenta. Si estamos trabajando en red es muy conveniente por claridad y seguridad que un mismo usuario tenga siempre en todas las máquinas el mismo login.

El segundo campo que nos encontraremos será el *password* del usuario encriptado. Esto es así ya que el fichero es visible por todos los usuarios y no se podría dejar sin codificar. El cambio de contenido se debe hacer con el comando ***passwd*** (el root puede utilizar este comando con un nombre de usuario para cambiar el *password* de alguien, pero tampoco sabrá el *password* que estaba asignado anteriormente). En entornos distribuidos que utilizan NIS (cuando desde varios computadores se accede a una misma cuenta residente en una única máquina) el comando adecuado será ***yppasswd***. Este password se comunicará al usuario de forma oral (no conviene que esté escrito en ningún sitio), el cuál deberá cambiarlo cunato antes.

A la hora de crear una cuenta siempre se deberá poner en este campo el símbolo "*" que hará que no se pueda acceder a la cuenta hasta que no se le asigne un password de la manera anteriormente indicada. Si dejamos el campo en blanco significa que ese usuario no tiene password, lo que constituye un agujero de seguridad tremendo.

Actualmente en la mayoría de sistemas, aparte del fichero `/etc/passwd` existe otro fichero donde están codificados los *passwords* de los usuarios, es el `/etc/shadow`, con la diferencia que ese fichero no es visible por los usuarios (excepto por el root). Esto es así porque teniendo un computador potente y las claves encriptadas de los usuarios se puede ejecutar un programa de desencriptación que las descubra a base de comparaciones con diccionarios en unas horas o días (para nombres de seis caracteres). Por eso aparece en el `/etc/passwd` en este campo una "x" y los password encriptados en `/etc/shadow` de forma no visible para "un vulgar mortal".

El siguiente campo es el número de usuario, como todas las entidades de UNIX están codificadas con un entero, en este caso con un número de 0 a 32767. Es conveniente que cuando estemos en un entorno distribuido usando NFS (el directorio personal [u otros] sólo reside en un disco de una máquina del sistema pero es visto por todas, a la hora de entrar a una máquina, ese directorio se monta por red y es visto desde allí) el número del usuario sea el mismo en todas las máquinas del sistema.

Existen unos cuantos números definidos para usuarios de instalación y el propio *root*, para usuarios creados por el administrador normalmente se empieza a contar a partir de 500, esto no es obligatorio, pero si lo es el no repetir ningún número para distintos usuarios. Los usuarios estándar aparecen junto con su información asociada en la siguiente tabla:

Usuario	UID	GID	Directorio raíz	Shell
root	0	0	`/root`	`/bin/bash`
bin	1	1	`/bin`	
daemon	2	2	`/sbin`	
adm	3	4	`/var/adm`	
lp	4	7	`/var/spool/lpd`	
sync	5	0	`/sbin`	`/bin/sync`

shutdown	6	0	/sbin	/sbin/shutdown
halt	7	0	/sbin	/sbin/halt
mail	8	12	/var/spool/mail	
news	9	13	/var/spool/news	
uucp	10	14	/var/spool/uucp	
operator	11	0	/root	
games	12	100	/usr/games	
gopher	13	30	/usr/lib/gopherdata	
ftp	14	50	/var/ftp	
nobody	99	99	/	

Igual que los ficheros se agrupan en directorios, los usuarios se concentran en grupos que también están codificados con un entero. El siguiente campo de `/etc/passwd` es ese número. Los grupos están localizados en algo similar al fichero `passwd`, ese fichero es `/etc/group`. Una tabla de grupos por defecto aparece en la tabla posterior.

Grupo	GID	Miembros
root	0	root
bin	1	root, bin, daemon
daemon	2	root, bin, daemon
sys	3	root, bin, adm
adm	4	root, adm, daemon
tty	5	
disk	6	root
lp	7	daemon, lp
mem	8	
kmem	9	
wheel	10	root
mail	12	mail
news	13	news
uucp	14	uucp
man	15	
games	20	
gopher	30	
dip	40	
ftp	50	
nobody	99	
usuarios	100	

Un campo opcional es el de los datos personales de contacto (*GECOS*), que suelen tener el nombre completo de la cuenta, la localización, el número de teléfono de trabajo y casa. Estos datos pueden ser obtenidos con el comando ***finger*** y pueden ser cambiados con ***chfn***.

Todos los usuarios (salvo raras excepciones que comparten *root*) tiene su propio directorio personal, normalmente colgado de `/home`. El directorio suele llamarse de la misma manera que la cuenta del usuario. Este directorio deberá tener los permisos y propietarios adecuados, esto lo conseguiremos con la combinación de comandos (para un supuesto usuario “rafa” que pertenece al grupo “usuarios”):

```
mkdir /home/rafa
chown rafa /home/rafa
chgrp usuarios /home/rafa
chmod 700 /home/rafa
```

que creará el directorio, le cambiará de dueño, de grupo y de modo de acceso, en este caso sólo con permisos para el dueño de la cuenta.

Por último, está el campo de la *shell*, donde indicaremos que tipo de *shell* arrancará el sistema cuando el usuario haga *login*. Todas las *shells* están en el directorio de binarios `/bin` y si no se especifica ninguna, el sistema arrancará la *shell* por defecto (Bourne) que es la ***sh***. Curiosamente no es obligado poner una *shell*, puede ser cualquier ejecutable, de hecho si se `/bin/false` es una manera de hacer que un usuario tenga prohibida la entrada.

Una vez que hemos creado la entrada en `/etc/passwd`, deberemos darle al usuario un *password* inicial. Esto se hace, como hemos mencionado, con el comando **`passwd`** seguido del nombre de usuario (cuidado de poner el nombre de usuario, si no ponemos nada cambiaríamos el de root). Una vez el usuario haya entrado al sistema lo podrá cambiar el mismo con ese comando (también se le pedirá el password antiguo), pero en este caso sin argumentos. Actualmente muchos de los sistemas por seguridad tienen comandos ***passwd*** que exigen introducir un *password* con ciertas características, como un determinado número de caracteres o una combinación adecuada de letras, números y signos de puntuación. Sino cumplimos con estas reglas (puestas para garantizar la seguridad del usuario) no podremos cambiar el *password* de la cuenta.

Si ya hemos creado el directorio del usuario como hemos indicado, el resto de tareas que deberemos de hacer no serán obligatorias, pero si bastante convenientes. La primera será copiar en ese directorio todos los ficheros de arranque necesarios, que dependerán de los programas que ejecute el usuario, incluida la shell. Una función del administrador es tener un directorio, por ejemplo `/usr/local/lib/arranque`, donde esté una colección adecuada de estos ficheros para ser copiados en el directorio de usuario (el sistema nos proporciona uno en el directorio `/etc/skel`), normalmente todos estos ficheros comenzarán por "." para que no sean visibles salvo con `ls -a`. Una vez hecho esto, podemos proceder así (advertir que no podemos hacer: `chown usuario /home/usuario/.*` ya que en ese directorio también está ".." que es `/home`, al que no podemos cambiar de usuario):

```
cp      /usr/local/lib/arranque/.[a-zA-Z]*   /home/usuario
chmod   644       /home/usuario/.[a-zA-Z]*
chown   usuario   /home/usuario/.[a-zA-Z]*
chgrp   grupo     /home/usuario/.[a-zA-Z]*
```

Otra tarea útil será configurar una cuenta de correo electrónico, sobre todo si estamos en un entorno distribuido para que el usuario reciba correo sólo en una máquina, para ello deberemos manipular el fichero `/etc/aliases` de forma conveniente:

```
usuario:      usuario@maquina
nombreapellido: usuario
```

haremos que el usuario reciba sólo el correo en la "maquina" y además tener un alias con el nombre y el apellido de alguna forma.

De forma muy opcional (sólo en sistemas antiguos) podremos actualizar el ficheros de grupos, que tendrá una línea por grupo con los siguientes campos (separados por ":"): nombre, password (vestigio nunca usado), numero de grupo GID, lista de usuarios del grupo separados por comas.

El resto de tareas dependerá de si el sistema lleva servicios de *accounting* (capítulo sexto) y cuotas (capítulo quinto). Por último sólo nos quedará entrar en la cuenta y verificarla. Se suele utilizar el comando **`pwd`** para ver si el directorio personal es el adecuado y `ls -la` para ver si están los ficheros de arranque.

Borrado, deshabilitación y modificación de usuarios

Cuando queramos que un usuario desaparezca del sistema deberemos asegurarnos de que se cumplan las siguientes tareas:

1. Si hay coutas de disco ponerlas a 0.
2. Quitar todos los alias que existan relacionados con él.
3. Matar todos sus procesos, si hay alguna corriendo, así como sus trabajos pendientes

(comando `at`) o tareas periódicas (`cron`).
4. Borrar todos los ficheros que estén en `/tmp o /vat/tmp` (pertenencientes a él).
5. Quitar la entrada en el fichero `/etc/passwd` y `/etc/group`.
6. Borrar su directorio `home`.
7. Quitar el *spool* de correo.

Existe un comando para borrar usuarios que es **`userdel`** (ver apartado siguiente).

Si lo que queremos es deshabilitar a un usuario temporalmente podemos tomar varias iniciativas:

1. Poner un "*" delante del password encriptado, de tal manera que el usuario no pueda entrar. Esto no es definitivo porque se podría todavía entrar desde la red.
2. Se pude cambiar la shell y poner un programa que pinte un mensaje advirtiendo del hecho. Este programa (*shell*) no debería estar en el fichero `/etc/shells` donde están las *shells* permitidas, para que no se pueda entrar en la cuenta, no sólo directamente, o con demonios como el de `telnet`, sino también por `ftp` o c correo (`sendmail`).
3. También se puede colocar el programa `/bin/false` para que no se haga nada.
4. También se puede utilizar el comando **`passwd`** con la opción -l (lock).

Por último, si queremos cambiar las características de un usuario, podemos hacerlo manualmente editando el `/etc/passwd` o usando el comando **`usermod`** (ver siguiente apartado).

Cambio de usuario

Si hemos entrado al sistema y queremos cambiar de usuario, podemos hacerlo con el comando **`login`**, al que podemos dar el nombre de usuario, si no es así, el propio comando nos lo pedirá. La utilización de `login` puede estar restringida creando el fichero `/etc/nologin` que será presentado en pantalla. Si el usuario al que queremos cambiar es `root`, primero se mirará el fichero `/etc/securetty` para ver desde que terminales podemos hacerlo, por ejemplo, sólo se podrá entrar como root desde el terminal principal, no de forma remota. También se pude restringir el uso de login a distintos tipos de usuarios en el fichero `/etc/usertty` (usar `man login` para más información).

Otra forma de cambiar de usuario es utilizar el comando `su` (sustituir). A diferencia del anterior, `su` crea una shell con un usuario distinto y por tanto podremos volver al original simplemente terminando la sesión.

Para evitar estos cambios en muchos sistemas (en particular en la distribución Ubuntu no existe el usuario root) existe la herramienta sudo (SUperuser DO) para realizar las tareas administrativas sin necesidad de cambiar al usuario root. El fundamento de sudo reside en su fichero de configuración, el fichero `/etc/sudoers`. Este fichero tiene, en los casos más sencillos, dos partes: una parte de alias y otra parte de reglas. La parte de alias, lo que hace es "agrupar" de alguna manera listas de usuarios y listas de aplicaciones. La parte de reglas define qué grupos de usuarios pueden usar qué grupos de programas con permisos distintos de los suyos y en qué condiciones pueden hacerlo.

6.2. Especial Red Hat

Gestión de Usuarios en RH

En este apartado veremos dos aspectos específicos de Red Hat, el tratamiento de los grupos de usuarios y las herramientas para shadow passwords.

En cuanto al primero surgió por la dificultad de encuadrar a los usuarios en los grupos de la manera tradicional, ya que en muchos proyectos, un usuario puede pertenecer eventualmente a varios grupos de trabajo y sin embargo él está asociado al grupo primario donde fue creado. Red Hat Linux utiliza un esquema de grupo privado de usuarios (UPG, del inglés "User Private Group") que permite que el manejo de grupos UNIX sea mucho más fácil. El esquema UPG no añade o cambia nada al modo tradicional UNIX de manejar internamente grupos, simplemente ofrece una nueva convención para ello, cada vez que se crea un nuevo usuario, por defecto, se le asigna un único grupo donde él es el único miembro, la máscara se cambia a 002 (antes era 022, ver comando `umask` que lee la máscara puesta a todo el sistema en el fichero `/etc/profile`) ya que ahora no es necesario extenderla al grupo, y el bit `setgid` (se hace con el comando `chmod g+s directorio`) para su directorio será activado. De esta manera, los archivos creados en ese directorio tendrán el grupo del directorio.

Veamos esto con un ejemplo, se tiene un proyecto llamado `proyecto` en el cual van a trabajar varios usuarios que pertenecen a sus propios grupos. Se creará un directorio para el mismo llamado `proyecto` y un grupo del mismo nombre. A ese directorio se le cambiara de grupo con el comando `chgrp`. Después se pone el bit `setgid` a ese directorio, con lo cual todos los ficheros creados en él pertenecerán al grupo del directorio y no del usuario que los cree. Por último se añaden los usuarios al nuevo grupo creado. Para hacer esto se cuenta con dos comandos:

- Para crear un grupo: `/usr/sbin/groupadd grupo`.
- Para añadir un usuario a un grupo: `/usr/bin/gpasswd -a usuario grupo`. Si quisieramos quitar a un usuario del grupo utilizaríamos la opción `-d`.

Si usáramos UPG y un usuario estuviera trabajando en muchos proyectos, no tendría que cambiar su máscara o grupo cuando se muevan de un proyecto a otro, ya que el `setgid` nos hace este trabajo.

En cuanto al segundo, el tratamiento de *shadow password*, existen varios comandos para trabajar con ellos, ya que manualmente no se puede editar el fichero `/etc/shadow`, ni se podría escribir el password encriptado, ya que el comando `passwd` trabaja exclusivament con el fichero `/etc/passwd`.

De todos modos, puede darse el caso de que no nos interese instalar esta utilidad:

1. Nuestro sistema no tiene cuentas de usuario.
2. Nuestro sistema está en red y las cuentas están por NIS(Network Information Services).
3. Existe en la máquina un software especial que revisa la entrada de usurios.

Si este no es el caso, será conveniente instalar esta utilidad (buscar paquete `shadow`), que entre otras cosas proporciona los siguientes comandos nuevos: `chage`, `newusers`, `dpasswd`, `gpasswd`, `useradd`, `userdel`, `usermod`, `groupadd`, `groupdel`, `groupmod`, `groups`, `pwck`, `grpck`, `lastlog`, `pwconv`, y `pwunconv`. Conviene destacar los siguientes:

- `pwconv`. Crea un fichero *shadow* a partir de un `/etc/passwd` y un virtual `/etc/shadow`. Primero, las entradas en el `shadow` que no estén en `passwd` son eliminadas; segundo, las entradas que no tienen "x" en `passwd` son añadidas al `shadow`; tercero, es añadido a `shadow` cualquier entrada nueva; y por último, los *password* en el fichero `passwd` son convertidos a "x". Para hacer la encriptación, se usa la librería `crypt`. Por seguridad, antes de efectuar la conversión se bloqueará los ficheros.
- `pwunconv`. Crea un fichero *passwd* desde un `/etc/passwd` y desde `/etc/shadow` y quita este último. Primero los *passwords* en el fichero `passwd` son añadidos desde el `shadow`; segundo, las entradas que existan en `passwd` pero no están en `shadow` son dejadas sueltas; y por último se borra el `shadow`. También se bloquean los ficheros.
- `pwck`. Verifica la integridad de sistema, es decir, todas las entradas en `/etc/passwd`

y `/etc/shadow`. Si alguna está mal (número de campos, nombre único, identificador de usurio y grupo válidos, correcto directorio home y *shell* adecuada), se preguntará al usuario si quiere borrarla.

6.3. Creación, Modificación y Borrado

Creación Automática, Modificación y Borrado

Toda esta tarea de creación que hemos visto en la sección anterior, puede ser llevada a cabo con una sola macro o comando que varía de un sistema a otro, pero que suele llamarse ***adduser*** o ***useradd*** y estar localizado `/usr/sbin`. La sintaxis del comando es:

```
useradd [-c comment] [-d home_dir]
        [-e expire_date] [-f inactive_time]
        [-g initial_group] [-G group[,...]]
        [-m [-k skeleton_dir] | -M] [-p passwd]
        [-s shell] [-u uid [ -o]] [-n] [-r] login
```

donde se puede dar toda la información necesaria para crear la cuenta, lo único imprescindible es el nombre de la cuenta.

Como vimos en el apartado anterior, otros comandos relacionados con la creación de cuentas de usuarios los podemos encontrar cuando se instala el paquete ***shadowutils***, las utilidades que se soportarán serán:

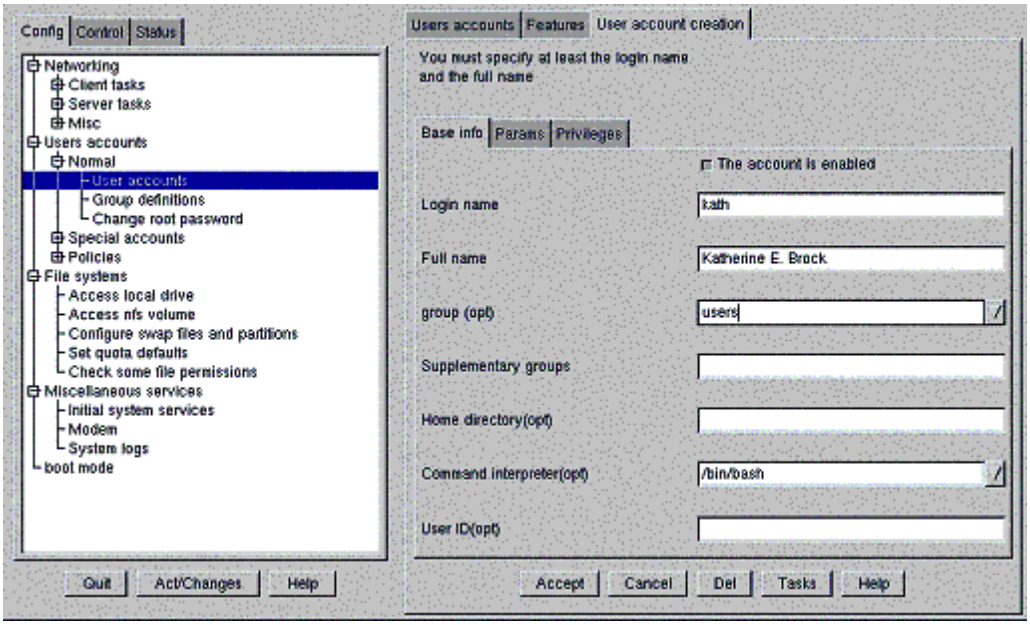
- Conversión de normal a ***shadow passwords*** y viceversa (***pwconv***, ***pwunconv***).
- Verificación de contraseñas, grupo y ficheros ***shadow*** asociados (***pwck***, ***grpck***).
- Métodos estándar en el mercado para añadir, borrar y modificar la cuenta de un usuario (***useradd***, ***usermod*** y ***userdel***).
- Métodos estándar en el mercado para añadir, borrar y modificar los grupos de un usuario (***groupadd***, ***groupmod*** y ***groupdel***).
- Métodos estándar en el mercado para administrar el fichero `/etc/group` (***gpasswd***).

Una vez que hemos creado una cuenta, podemos tener intención de modificarla o borrarla, como hemos visto, manualmente, pero también existen comandos para hacerlo de un solo golpe, la sintaxis de los comandos para modificar la cuenta de un usuario o borrarla es:

```
usermod [-c comment] [-d home_dir [ -m]]
        [-e expire_date] [-f inactive_time]
        [-g initial_group] [-G group[,...]]
        [-l login_name] [-p passwd]
        [-s shell] [-u uid [ -o]] [-L|-U] login

userdel [-r] login
```

Si estamos usando un terminal gráfico (también se puede hacer en uno de texto pero con otra interfase más primitiva) en el sistema Red Hat podemos utilizar la aplicación Linuxconf (la aplicación es tan avanzada que incluso se puede usar por web a través del puerto 98 de esta manera: http://el.nombre.de.la.maquina:98, antes habrá que configurar la máquina LINUX para que acepte este tipo de accesos desde el cliente) de código abierto , que nos permitirá cambiar, modificar o borrar usuarios (cuentas). En lo que se refiere a la creación de cuentas de usuario su aspecto sería:



6.4. Auditoría y Contabilidad

Auditoría y Contabilidad

En todos los sistemas modernos el sistema operativo lleva un control (accounting) sobre la actividad del sistema generando ficheros de contabilidad (log), en particular esto se hace también para los usuarios del sistema. Estos ficheros suelen estar residentes en los directorios `/var/log` y `/etc`. Normalmente esta información es muy prolija y se recoge, visualiza o utiliza a través de comandos específicos, por ejemplo los comandos vistos who y finger. Los responsables de estos ficheros suelen ser demonios que se controlan a través del fichero `/etc/syslog.conf`.

Los ficheros más importantes dentro de `/var/log` son los siguientes:

- **messages**. Son los mensajes que aparecen en la pantalla del administrador, de tal manera que ninguno de estos mensajes se pierda.
- **lastlog**. En este fichero se escribe una entrada cada vez que se produce un login en el sistema, en algunos sistemas puede ser visto directamente, en otros está codificado. Para ver ver este fichero bien por usuarios o por días se utiliza el comando **`lastlog`**.
- **utmp**. Proporcionan información sobre los usuarios conectados, también puede estar en `/var/run` y se utiliza normalmente con el comando **`who`**.
- **wtmp**. Proporcionan información sobre la duración de las sesiones, se consulta normalmente con los comandos **`last`** y **`who`**.
- **btmp**. Proporcionan información sobre los intentos fallidos de conexión.

Estos ficheros pueden llegar a ser gigantescos por lo que para tratarlos se utiliza el comando **`logrotate`** que puede hacer del fichero un buffer circular, almacenarlo comprimirlo o mandarlo incluso por correo. También se puede activar por tiempo o por tamaño del fichero (ver cron en apartado posterior).

Además de lo explicado, en UNIX / GNULinux también se puede llevar una contabilidad de los recursos

utilizados (a partir del núcleo 1.3.73), desgraciadamente la gestión de la contabilidad depende del sistema y además las herramientas no suelen estar instaladas. Hay dos sistemas fundamentales: el sistema BSD (Berkeley) y el de System V, aunque se suele emplear mucho menos. Los pasos para activar la contabilidad suelen ser:

1. Traer el paquete adecuado e instalarlo. El nombre es: quota-acct.
2. Modificar el system init script para que habilite la contabilidad con el comando **`/sbin/accton`** en el fichero `/var/log/pacct`.
3. Crear ese fichero si es necesario con **`touch`**. El fichero debe pertenecer a root con permisos de lectura/escritura y sólo lectura para el resto.
4. Rebotar la máquina para que tenga efecto.

Los comandos para trabajar después serán:

- **`ac`**
  Sirve para hacer algo similar a **`last`** con la ventaja de que puede mostrar los tiempos totales por día (-d) y por cada usuario (-p).
- **`accton`**
  Sirve para apagar o encender el proceso de contabilidad, normalmente es colocado en las macros de inicio.
- **`sa`**
  Resume (summarizes accounting) información de comandos previamente ejecutados y que han sido registrados en `/var/account/pacct`.
- **`lastcomm`**
  Igual que el anterior pero muestra toda la información.

6.5. Comunicación Usuarios

El último punto que nos queda por ver en este capítulo sobre usuarios, es la comunicación que podemos establecer con ellos. Existen varios mecanismos que podemos utilizar:

1. **message of the day**. Es un fichero que está en `/etc/motd` y que se presentará en pantalla después que un usuario ha realizado un ***login*** con éxito. Es un mecanismo que consume muchos menos recursos que mandar un mail a todos los usuarios.

2. ***wall***. Si queremos mandar un mensaje a todos los usuarios que estén en ese momento conectados, podremos usar el comando ***wall***. Este comando aceptará como argumento un mensaje terminado en EOF (aunque no es necesario) ya que esta pensado para redirección de entrada. Para que los mensajes puedan llegar a los usuarios, estos deben tener habilitado el terminal para escritura con la orden ***mesg*** que toma un argumento con dos valores: **y** o **n**, aunque el *root* siempre tiene este permiso.

3. ***write***. Idéntico al anterior, pero en este caso sólo se manda un mensaje al usuario indicado como primer argumento, en el segundo podemos poner opcionalmente el terminal donde está conectado, esto tendrá sentido si el usuario está en varios terminales. El mensaje se dará a continuación de forma interactiva terminado en EOF (ctrl.−d desde el teclado).

4. ***talk***. Este comando tiene el mismo propósito que el anterior pero lo hace de una forma "visual", dividiendo la pantalla en dos, en una parte aparecen los mensajes enviados y en otra los recibidos. Toma los mismos argumentos que ***write***, pero en este caso el usuario también puede estar en una máquina remota tomando la forma: `usuario@maquina`. Este comando hace uso de un demonio llamado `talkd` para establecer las conexiones.

5. ***mail***. Si lo que queremos es mandar un aviso a un usuario que no esté conectado, no nos quedará otro remedio que hacerlo con el tradicional ***mail***. Para hacer esto deberemos poner como argumento el nombre del usuario(s) (se puede mandar una copia con la opción `-c`) al que queremos mandar el correo y después el mensaje en si terminado en EOF (si queremos poner un asunto [*subject*] al mensaje lo podremos hacer con la opción `-s`). Si lo que queremos es leer el correo bastará con que utilicemos el comando sin argumentos, después podemos utilizar distintos comandos interactivos para actuar sobre los mensajes recibidos: ***d*** para borrar, ***r*** para responder o ***x*** para abortar una sesión o ***q*** para salir.

   Hay varios ficheros relacionados con el correo:

   - `/var/spool/mail/usuario`. El fichero general del usuario para correo.
   - `~/mbox`. Fichero de correo de mensajes obsoletos.
   - `~/.mailrc`. Servirá para introducir órdenes al mail, como por ejemplo, hacer listas de correo con ***alias*** (también se puede hacer alias de correo con el fichero `/etc/aliases`).
   - `/tmp/R*`. Ficheros de correo temporales.
   - `/usr/lib/mail.help`. Ficheros de ayuda.
   - `/etc/mail.rc`. Fichero de inicialización.