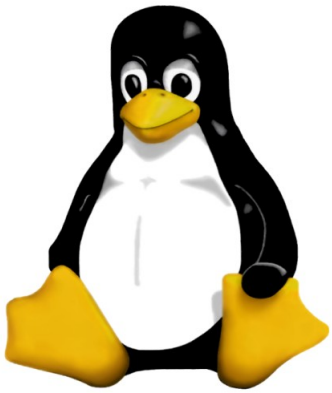
GNU/Linux

Última modificación 2009/02a



2008-2009 – Güimi (http://guimi.net)
Esta obra está bajo una licencia "Reconocimiento-Compartir bajo la misma licencia 3.0 España" de Creative Commons.
Para ver una copia de esta licencia, visite http://guimi.net/index.php?pag_id=licencia/cc-by-sa-30-es_human.html.

Reconocimiento tautológico: Todas las marcas pertenecen a sus respectivos propietarios.

Fuentes principales: "Unix Toolbox" de Colin Barschel (http://cb.vu/unixtoolbox.xhtml), Wikipedia (http://www.wikipedia.org) y Debian (http://www.debian.org).
Algunas partes son directamente copia o traducción de las fuentes.

GNU/Linux

Contenido

INTÉRPRETES DE COMANDOS

INTRODUCCIÓN

El primer intérprete de comandos de UNIX fue **sh** (*shell*) o "Thompson Shell", más tarde sustituido por una versión de Bourne del mismo nombre y conocida como "Bourne Shell". Aunque pretendía ser un intérprete de comandos interactivo, ganó popularidad como un lenguaje de scripting.

En los 70s Bill Joy trabajaba para la versión de UNIX de Berkeley y desarrolló **csh**[1], la primera alternativa al Bourne Shell, pero con la sintaxis del lenguaje C -y por tanto incompatible con **sh**- y mucho más avanzado. Incorporaba historia de comandos, gestión de trabajos, autocompletado de nombres...
Su potencia fue una de las claves para la amplia aceptación de los UNIX BSD.
Una versión más moderna de **csh**, llamada **tcsh** (Tenex csh), es el intérprete por omisión de los UNIX derivados de BSD, incluido Mac OS X (hasta la versión 10.4 en que se usa bash).

Como respuesta a **csh** David Korn desarrolló a principio de los 80 para AT&T el Korn Shell, llamado **ksh**, como un intérprete compatible con **sh** pero incluyendo muchos elementos de **csh**, como un historial de órdenes editables de forma compatible con **vi** o con **emacs**. Una de las bazas que jugó en contra de su aceptación fue el echo de que se distribuyese de manera comercial como software cerrado, haciendo que los scripts en **ksh** no tuvieran garantizada su portabilidad. Éste es el intérprete por omisión de los UNIX comerciales evolucionados desde el de AT&T y el estándar POSIX.

En 1987 Brian Fox creó el "Bourne-again shell[2]" o **bash**, como software libre compatible con **sh**, añadiendo las mejores características de **ksh** y **csh**. De hecho cuando se invoca a **sh** en un sistema GNU/Linux, en realidad se está ejecutando **bash** en modo de compatibilidad total. Al ser libre además de ser el intérprete predefinido de GNU/Linux y estar disponible para otros sistemas ha evolucionado a gran velocidad hasta convertirse en el intérprete más popular.

En todo caso cuando se quiere garantizar la portabilidad total de un script se sigue utilizando **sh**.
Para que un script utilice un intérprete u otro, se indica en la primera línea del mismo. Ejemplos:

```
#!/bin/bash
#!/bin/csh -f
#!/bin/perl
```

Se puede utilizar el intérprete deseado como cualquier otro comando, por ejemplo:

```
ksh
bash script
. script
```

El indicador (*prompt*) de los intérpretes de comandos se cierra habitualmente con '#' para root y para los usuarios se usa principalmente '$' (bash, ash, sh) y '%' (csh, tcsh, zsh).

Se puede invocar un "subshell" dentro de un comando usando `` `comando` `` o `$(comando)` . La segunda opción es preferible pero solo está disponible en los **bash** modernos. Por ejemplo:

```
sudo aptitude install linux-headers-$(uname -r)
```

1 Además de dirigir Unix BSD, desarrolló vi y NFS. También es co-fundador de Sun Microsystems.
2 Juego de palabras entre "Otra vez bourne shell" y "Shell renacido (born-again)".

USO BÁSICO DEL INTÉRPRETE

Algunas combinaciones de teclas de uso común en los intérpretes (parte pueden modificarse con **`stty`**):

- Ctrl-C: Envía la señal de interrupción al proceso en ejecución.
- Ctrl-D: Indica el final de un flujo de datos.
- Ctrl-Z: Pasa el proceso activo a "*background*".
- Alt+F1...F12: Cambio de consola a tty1...tty12

Algunas combinaciones de teclas interesantes en **`bash`**.

- Ctrl-r Búsqueda de comandos en el histórico
- Ctrl-l Limpia la terminal (equivalente a clear)
- Ctrl-k Corta caracteres desde el cursor hasta el final
- Ctrl-u Corta caracteres desde el cursor hasta el inicio
- Ctrl-w Borra la palabra que está antes del cursor
- Ctrl-y Pega el texto que fue cortado, a partir de la posición del cursor
- Tab Completa comandos o rutas, según existan
- Tab-Tab Muestra opciones que cumplen con el inicio de un patrón

Completando comandos:

```
$ ls /bin/m[tab][tab]
mkdir mkfifo mknod more mount mv
$ ls /bin/mor[tab]
$ ls /bin/more
```

Histórico de comandos

Las flechas de cursor sirven para moverse por el histórico de comandos.

El comando **`history`** muestra un historial enumerado de los últimos comandos.

```
$ history
 1 cd /home/
 2 cd /mnt/cdrom/
 3 ls
 4 history
```

Para hacer uso de alguno de los comandos se utiliza el signo de admiración (!) seguido del número del comando.

```
$ !3
```

Ejecución de varios comandos

Para ejecutar más de un comando en la línea de entrada, basta con separar las órdenes con punto y coma (;). También se puede hacer condicionalmente con "&&", lo que hará que solo se ejecute el siguiente comando si el anterior ha terminado correctamente. Ejemplos:

```
$ clear; logout
$ sudo aptitude update && sudo aptitude upgrade
```

Redirección de salidas y entradas. Tuberías

Se puede utilice el operador '>' para redireccionar la salida estándar de un comando a un archivo sobreescribiendo su contenido, o '>>' para que lo añada al final del mismo. Si se indica '2>' o '2>>' redirigimos la salida de errores. También podemos redirigir la salida de errores a la salida estándar (2>&1) y viceversa (>&2).
También puede redirigirse la entrada con '<'.

Por último pueden generarse tuberías con '|'. Esto redirige la salida de un proceso a la entrada del siguiente.

```
mount | cut -d ' ' -f1,3,5-6 | column -t     # Muestra FS montados | se 'embellece' la salida
```

Background/Foreground

Un proceso puede pasarse al fondo (*background*) o traer a primer plano (*foreground*) con Ctrl-Z, **`bg`** y **`fg`**.

```
# jobs -l                                   # List processes in background
[1] - 36232 Running                               ping cb.vu > ping.log
[2] + 36233 Suspended (tty output)                top
# fg %2                                     # Bring process 2 back in foreground
```

Se puede utilizar **`nohup`** para mantener un proceso en ejecución aunque se cierre el shell.

```
# nohup ping -i 60 > ping.log &
```

Prioridad de los procesos

Los comandos **`nice / renice`** indican lo amable que es un proceso. Cuanto más amable es, menos prioridad tiene [-20 - 20]. Solo **`root`** puede adjudicar valores de **`nice`** negativos (muy prioritarios).

```
# nice -n 5 top                             # Weaker priority (/usr/bin/nice)
# renice -5 586                             # Stronger priority for running process 586
```

Entrecomillado

Cuando se usan comillas dobles el intérprete sustituye las variables por su contenido, con comillas simples no. Además unas comillas permiten el uso de las otras como caracteres normales.

```
$ echo "La variable path es $PATH"
La variable path es /usr/local/bin:/usr/bin:/bin:/usr/bin/X11:/usr/games:/home/guimi/bin
$ echo 'La variable path es $PATH'
La variable path es $PATH
$ echo 'La variable path es "$PATH"'
La variable path es "$PATH"
$ echo "La variable path es '$PATH'"
La variable path es '/usr/local/bin:/usr/bin:/bin:/usr/bin/X11:/usr/games:/home/guimi/bin'
```

Símbolos del indicador (*prompt*)

```
$ echo "'$PS1' -- '$PS2'"
'@\[\]\h\[\]:${newPWD}\$ ' -- '> '
```

Interacción con la terminal / Parámetros de la terminal (echo, read, clear, reset, stty, read)

```
clear                 # Limpia la terminal
reset                 # Reestablece los parámetros básicos de la terminal
stty -a               # Consulta todos los parámetros de la terminal en modo legible por humanos
set | grep $USER      # lista las variables de entorno
```

```
setenv varname value                        # Set env. variable varname to value (csh/tcsh)
export varname="value"                      # set env. variable varname to value (sh/ksh/bash)
```

```
echo -n "Clave:"       # Mostramos el mensaje "Clave:" sin realizar un salto de línea
oldmodes=`stty -g`     # Tomamos nota del estado de stty
stty -echo             # Eliminamos el echo local (por ejemplo para pedir una clave)
read clave             # Leemos clave de la entrada estándar
stty $oldmodes         # Volvemos a situar stty como estaba
exit                   # Cierra la sesión (igual que Ctrl+d)
```

COMANDOS BÁSICOS

Manual (man, whatis, apropos, info)

```
man man        # Muestra el manual del comando man
whatis whatis  # Muestra la descripción del comando en el manual
apropos pdf    # Muestra los comandos que en su descripción en el manual aparece "pdf"
info info      # Sistema de información de GNU
```

Listados (ls)

```
ls -alFq       # Opciones: todos, lista larga, con indicación de tipo, indicando carac. no gráficos
ls -Rsh        # Opciones: recursivamente, ordenados por tamaño, tam. legible por humanos
ls -lid directorio    # Opciones: muestra inodos,  lista directorios no sus contenidos
ls -tr         # Opciones: ordena por marca de tiempo, invierte el orden (más reciente al final)
```

Directorios (cd, mkdir, rmdir)

```
cd directorio  # Cambia al directorio indicado (si no se indica cambia al directorio $HOME)
mkdir -p directorio/subdirectorio    # Crea el directorio indicado y sus antecesores
rmdir -p directorio/subdirectorio    # Borra el directorio indicado y sus antecesores
```

Ficheros (touch, rm, mv, cp, file, ln)

```
touch fich1           # Modifica la fecha de acceso al fichero y lo crea si no existe
mv [-i|-f] dir1 dir2  # Mueve de forma interactiva o forzosa
cp -Rp [-i|-f] d1 d2  # Copia recursiv. y manteniendo permisos de forma interac. o forzosa
rm -rf directorio     # Borrar recursivamente y sin pedir confirmación
file fich1            # Indica el tipo de fichero de fich1
ln [-s] fich1 fich2   #  Crea un enlace [simbólico] de fich1 llamado fich2
                              (no se permiten enlaces duros de directorios)
```

Contenido de ficheros (cat, head, tail, more, less)

```
cat [-n] fich*        # Vuelca el contenido de fich* [indicando el número de línea]
head [-n 25] fich1    # Vuelca las primeras 10 [25] líneas de fich1
tail [-n 25] [-f] log # Vuelca las ultimas 10 [25] líneas de log [y añade según se crean]
```

Filtros (sort, uniq, cut, tr, wc, column)

```
sort [-f] [-r] fich1  # Ordena [ignorando may/min] [invertido] fich1
uniq [-c] [-d] fich1  # Elimina las lineas repetidas consecutivas de fich1
                              [incluyendo la cuenta] [muestra solo líneas con alguna repetición]
uniq [-u] [-i] fich1  # [muestra solo las líneas sin repeticiones] [ignora may/min]
cut -c1-2 fich1       # Muestra los dos primeros caracteres de cada línea de fich1
tr [-d 'x'] [-s ' '] '|' ','  # traduce de la entrada estándar '|' por ','
                                    [elimina 'x'] [elimina repeticiones de ' ']
wc [-l|-w|-c] fich1   # Cuenta [solo lineas|palabras|caracteres]
column -t             # Muestra la entrada de forma tabulada
```

```
cat fich* | sort | uniq | wc -l      # Vuelca fich*, ordena el volcado, elimina las lineas repetidas
                                      y lo muestra paginado
wc -l fich1 | cut -d ' ' -f1  # Cuenta lineas fich1 | Muestra campo 1 (usa ' ' como separador)
```

Fechas y tiempos (date, cal, time)

```
cal 9 1752            # Muestra el calendario de septiembre de 1752
date +%C%y/%m/%d-%X   # Muestra la fecha en formato Ccyy/mm/dd/-HH:MM
date 123123592008     # Establece la fecha y hora 2008/12/31 23:59
time programa         # Muestra el consumo de tiempo de la ejecución de programa
ntpdate -u hora.uv.es # Establece la hora del sistema
```

Permisos y propiedad (chmod, chown, chgrp, umask, lsattr, chattr)

La máscara reside en **`/etc/profile`**, modificable con **`umask`**. Usualmente es 022, por lo que otorga permisos de 755.

```
chmod 640 fich1                      # Restringe fich1 a -rw-r-----
chmod u=rw,g=r,o= fich1              # Igual que lo anterior
chmod -R a-x /home/*                 # Recursivamente quita x en las tres ternas
chmod u+s fich1                      # Establece el SetUID para el usuario
chown usuario fich1                  # Modifica el usuario propietario de fich1
chgrp grupo fich1                    # Modifica el grupo propietario de fich1
chown usuario:grupo fich1            # Modifica el usuario y grupo propietarios de fich1
chmod 751 $(find ./ -type d -print)  # Establece los permisos 751 para los subdirectorios
umask 022                            # Establece la máscara de creación a 022
```

Gestión de usuarios (group/user add/del, passwd, chage)

```
# groupadd admin                    #  Crea el grupo "admin"
# useradd / adduser -c "Guimi" -g admin -m guimi   #  Crea el usuario "guimi"
# userdel / deluser guimi           #  Borra el usuario "guimi"
# passwd [usuario]                  # Cambia la clave [de "usuario" -solo root-]
# chage -I                          # Cambia -interactivamente- datos de expiracion de clave
```

Los datos de usuario se encuentran en **`/etc/passwd`**, las claves cifradas se encuentran en **`/etc/shadown`** y los datos de grupos en **`/etc/group`**.
Para evitar temporalmente el acceso de usuarios basta crear el archivo **`/etc/nologin`**, cuyo contenido se mostrará a quién intente acceder.

Búsqueda de ficheros (find, locate, updatedb)

```
find / -perm -u+s -print      # Busca todos los programas con el bit SUID
find . -type d -exec chmod a+x \{\} \;      # Establece permisos a+x para todos los subdirectorios
find . -iname guimi -xdev     # Lista ficheros "*guimi*" [ignorando may/min] sin cambiar de fs
find   /home/user/ -cmin 10 -print   # Ficheros creados hace menos de 10 minutos
find . -size +10M -size -50M -print  # Ficheros con tamaño entre 10 MB y 50 MB
```

```
locate guimi                  # Lista ficheros con "guimi" en el nombre de su base de datos
updatedb                      # genera la base de datos de locate
```

Información (uname, which, pwd, id, who, last, uptime, hostname, dmesg)

```
uname -a              # Muestra información sobre el kernel
uname -r              # Muestra la versión del kernel
which comando         # Indica la ruta del primer ejecutable "comando" en la ruta $PATH
pwd                   # Indica la ruta del directorio en que nos encontramos
id [usuario]          # Muestra información de UID y GIDs de "usuario" o de quien ejecuta el comando
who [usuario]         # Muestra información sobre las conexiones de usuarios (o solo de "usuario")
last [usuario]        # Muestra información sobre las últimas conexiones de usuarios
last reboot           # Muestra información sobre los últimos reboot
uptime                # Muestra el tiempo que lleva encendido el equipo y su carga
hostname              # Muestra el nombre del equipo
hostname -i           # Muestra la dirección IP del equipo
dmesg                 # Muestra y controla el buffer del kernel
lsb_release -a        # Muestra toda la información LSB de la publicación de la distro
```

```
cat /etc/debian_version       # Muestra la versión de Debian
cat /etc/XXXX-release         # Versión de SuSE, lsb (Ubuntu), redhat, gentoo, mandrake, sun...
```

Diferencia de ficheros / Parches (diff, patch)

```
diff {-u|-c} fich1 fich1.mod > fich1.patch  # Creas un parche del tipo usual o más estándar
patch -b fich1 < fich1.patch                # Aplica un parche guardando copia de fich1
```

Filtros avanzados (grep, egrep, fgrep, xargs, gawk, sed)

```
grep [-A 2] [-B 3] guimi fich1 # Muestra líneas con "guimi" en fich1 [+2 lín. desp.] [+3 lín. antes]
  [-i] [-R] [-v]               # [ignorando may/min] [recursivamente] [lín. SIN "guimi"]
  [-e "regexp"] [-E "extended-regexp"]
```

`egrep` equivale a **`grep -E`**, **`fgrep`** equivale a **`grep -F`** y **`rgrep`** equivale a **`grep -R`**.
`pgrep` equivale a mostrar los pid resultantes de "**`ps aux | grep`**".

COMANDOS AVANZADOS

Información sobre *Hardware*

```
# dmesg                                  # Detected hardware and boot messages
# lsdev                                  # information about installed hardware
# dd if=/dev/mem bs=1k skip=768 count=256 2>/dev/null | strings -n 8 # Read BIOS
# cat /proc/cpuinfo                      # CPU model
# cat /proc/meminfo                      # Hardware memory
# grep MemTotal /proc/meminfo            # Display the physical memory
# watch -n1 'cat /proc/interrupts'       # Watch changeable interrupts continuously
# free -m                                # Used and free memory (-m for MB)
# cat /proc/devices                      # Configured devices
# lspci -tv                              # Show PCI devices
# lsusb -tv                              # Show USB devices
# lshal                                  # Show a list of all devices with their properties
# dmidecode                              # Show DMI/SMBIOS: hw info from the BIOS
```

Información sobre dispositivos de almacenamiento

```
# hdparm -I /dev/sda                      # information   about the IDE/ATA disk (Linux)
# fdisk /dev/ad2                          # Display and   manipulate the partition table
# smartctl -a /dev/ad2                    # Display the   disk SMART info
# df                                      # display free disk space and mounted devices
# cat /proc/partitions                    # Show all registered partitions (Linux)
# du  -sh *                               # Directory sizes as listing
# du  -csh                                # Total directory size of the current directory
# du  -ks * | sort -n -r                  # Sort everything by size in kilobytes
# ls  -lSr                                # Show files, biggest last
```

Estadísticas, mensajes y carga del sistema

```
# top                                    # display and update the top cpu processes
# vmstat [2]                             # display virtual memory statistics (2s intervals)
# tail -n 500 /var/log/messages          # Last 500 kernel/syslog messages
# tail /var/log/warn                     # System warnings messages see syslog.conf
```

Límites

De un shell o script

```
# ulimit -a                              # Muestra los límites del usuario
# ulimit -n 1024                         # Modifica el límite de archivos abiertos
```

De usuarios y procesos en **`/etc/security/limits.conf`**.

```
*    hard    nproc    1000
```

Del sistema en **`/etc/sysctl.conf`** y mediante **`sysctl`**.

```
# sysctl -a                              # View all system limits
# sysctl fs.file-max                     # View max open files limit
# sysctl fs.file-max=102400              # Change max open files limit
```

Módulos del núcleo

```
# lsmod                                   # List all modules loaded in the kernel
# modprobe isdn                           # To load a module (here isdn)
```

Compilación del núcleo

```
# cd /usr/src/linux
# make mrproper                           # Clean everything, including config files
# make oldconfig                          # Reuse the old .config if existent
# make menuconfig                         # or xconfig (Qt) or gconfig (GTK)
# make                                    # Create a compressed kernel image
# make modules                            # Compile the modules
# make modules_install                    # Install the modules
# make install                            # Install the kernel
# reboot
```

Listado de procesos y PIDs

Cada proceso tiene un identificador único (PID). Se pueden listar con ps:

```
# ps aux                                  # Extensive list of all running process
# ps aux | grep cron                      # However more typical usage is with a pipe
# pgrep -l sshd                           # Find the PIDs of processes by (part of) name
# echo $$                                 # The PID of your shell
# fuser -va 22/tcp                        # List processes using port 22 (Linux)
# fuser -va /home                         # List processes accessing the /home partiton
# strace df                               # Trace system calls and signals
# history | tail -50                      # Display the last 50 used commands
```

top

El comando top muestra información de los procesos en ejecución y su consumo de recursos. Durante su ejecución se puede pulsar h para obtener ayuda o, entre otras opciones:

- u [user name] To display only the processes belonging to the user. Use + or blank to see all users
- k [pid] Kill the process with pid.
- 1 To display all processors statistics (Linux only)
- R Toggle normal/reverse sort.

kill

Permite enviar señales (no solo KILL) a un proceso.

```
# kill -s TERM 4712                       # same  as kill -15 4712
# killall -1 httpd                        # Kill  HUP processes by exact name
# pkill -9 http                           # Kill  TERM processes by (part of) name
# pkill -TERM -u www                      # Kill  TERM processes owned by www
# fuser -k -TERM -m /home                 # Kill  every process accessing /home (to umount)
```

Las señales más importantes son:

1	HUP (hang up)
2	INT (interrupt)
3	QUIT (quit)
9	KILL (non-catchable, non-ignorable kill)
15	TERM (software termination signal)

rsync

`rsync` puede casi reemplazar completamente a **`cp`** y **`scp`**. Además permite recuperar eficientemente una transmisión interrumpida. Tradicionalmente para funcionar en red utiliza un servidor de rsyn, pero hoy en día su uso principal es sobre SSH -más lento, pero más seguro-. Algunos ejemplos:

Copiar los directorios y su contenido completo:

```
# rsync -a /home/colin/ /backup/colin/
# rsync -aR --delete-during /home/user/ /backup/   # Usa paths relativos y borra durante no antes
```

Lo mismo pero en red y comprimiendo:

```
# rsync -axSRzv /home/user/ user@server:/backup/user/
```

Usando el puerto 20022 para la conexión SSH:

```
# rsync -az -e 'ssh -p 20022' /home/colin/ user@server:/backup/colin/
```

Ejemplos usando el demonio rsync ("::"). Copia y restauración:

```
# rsync -axSRz /home/ ruser@hostname::rmodule/backup/
# rsync -axSRz ruser@hostname::rmodule/backup/ /home/               # To copy back
```

Algunas opciones interesantes:

-a, --archive	archive mode; same as -rlptgoD (no -H)
-r, --recursive	recurse into directories
-R, --relative	use relative path names
-H, --hard-links	preserve hard links
-S, --sparse	handle sparse files efficiently
-x, --one-file-system	don't cross file system boundaries
--exclude=PATTERN	exclude files matching PATTERN
--delete-during	receiver deletes during xfer, not before
--delete-after	receiver deletes after transfer, not before

sudo

Autorizamos algunos usuarios a realizar tareas administrativas

```
# vi /etc/sudoers
# DEFAULTS
# env_reset : resetea las variables de entorno antes de ejecutar
# insults : cuando te equivocas de contraseña
#+    te "insulta" irónicamente
# timestamp_timeout=5 : una vez pones la contraseña correctamente,
#+    no te la vuelve a pedir en 5 minutos (0 : pedir siempre)
# lecture=once : solo te avisa sobre la responsabilidad una vez por sesión
# passwd_tries=2 : permite 2 intentos para la contraseña
# rootpw : pide la clave de root en vez de la del usuario

Defaults env_reset, insults, timestamp_timeout=5, lecture=once, passwd_tries=2

# User alias specification
User_Alias ADMINS=guimi

# Cmnd alias specification
Cmnd_Alias APT = /usr/bin/apt-get, /usr/bin/dpkg, /usr/bin/aptitude
Cmnd_Alias APAGADO = /sbin/shutdown, /sbin/halt, /sbin/reboot
Cmnd_Alias RED = /usr/bin/kismet, /sbin/iwlist, /sbin/ifuop, /sbin/ifdown, /sbin/wpa_cli
Cmnd_Alias PROPIOS = /root/bin/SCRIPT1

# User privilege specification
root    ALL=(ALL) ALL
# Si el usuario admin esta en el grupo 'sudo' no le pedirá clave
admin   ALL=(ALL) ALL

# Indicamos expresamente que no solicite clave para estos comandos
ADMINS ALL= NOPASSWD: APT
ADMINS ALL= NOPASSWD: APAGADO
ADMINS ALL= NOPASSWD: RED
ADMINS ALL= NOPASSWD: PROPIOS
```

```
# sudo /etc/init.d/dhcpd restart                # Run the rc script as root
# sudo -u sysadmin whoami                       # Run cmd as an other user
```

screen

Tiene dos funcionalidades principales:

- Permitir múltiples sesiones de terminal en una sola terminal (o conexión).
- Desacoplar los programas en ejecución de la terminal real para que sigan en funcionamiento en background aunque se cierre la terminal, que podrá reconectarse más tarde.

```
 # screen
```

Dentro de la sesión ejecutamos un programa, por ejemplo **`top`**, y lo desacoplamos con Ctrl-a Ctrl-d.
Desde otra sesión de terminal (o la misma) reacoplamos el proceso con **`screen -r`** o **`screen -R -D`**.

Comandos (dentro de **`screen`**). Todos empiezan por Ctrl-a:

- Ctrl-a ? help and summary of functions
- Ctrl-a c create an new window (terminal)
- Ctrl-a Ctrl-n and Ctrl-a Ctrl-p to switch to the next or previous window
- Ctrl-a Ctrl-N switch to windows N [0-9]
- Ctrl-a " to get a navigable list of running windows
- Ctrl-a a to clear a missed Ctrl-a
- Ctrl-a Ctrl-d to disconnect and leave the session running in the background
- Ctrl-a x lock the screen terminal with a password

dd

El programa dd (disk dump) permite copiar particiones y discos y realizar pequeños trucos al respecto.

```
# dd if=<source> of=<target> bs=<byte size> count=x conv=<conversion>
```

El tamaño de bloque (bs) predefinido es 512 (un bloque). Se puede usar tamaños mayores pero consume más memoria.

```
#  dd if=/dev/hda of=/dev/hdc bs=16065b                        # Copy disk to disk (same size)
#  dd if=/dev/sda7 of=/home/root.img bs=4096 conv=notrunc,noerror # Backup /
#  dd if=/home/root.img of=/dev/sda7 bs=4096 conv=notrunc,noerror # Restore /
#  dd bs=1M if=/dev/sda | gzip -c > sda.gz                      # Zip the backup
#  gunzip -dc sda.gz | dd of=/dev/sda bs=1M                     # Restore the zip
#  dd bs=1M if=/dev/sda | gzip | ssh eedcoba@fry 'dd of=sda.gz' # also remote
#  dd if=/dev/sda1 of=/dev/sdb1 skip=1 seek=1 bs=4k conv=noerror             # Skip MBR
     # This is necessary if the destination (sdb1) is smaller.
```

```
# dd if=/dev/hda of=/dev/hda                                  # Refresh the magnetic state
   # The above is useful to refresh a disk. It is perfectly safe, but must be unmounted.
```

Trucos

```
# dd if=/dev/zero of=/dev/hdc                                 # Delete full disk
# dd if=/dev/urandom of=/dev/hdc                              # Delete full disk better
```

```
time dd if=/dev/hda1 of=/dev/null bs=1024k count=1000      # rendimiento de lectura
time dd if=/dev/zero bs=1024k count=1000 of=/home/1Gb.file # rendimiento de escritura
```

```
# dd if=/dev/mem bs=1k skip=768 count=256 2>/dev/null | strings -n 8 # Read BIOS
```

El registro MBR está en el primer bloque (512 B), estando los primeros 63 libres. Los primeros 446 son el "boot loader" o cargador de arranque y el resto son la tabla de particiones.

```
#  dd  if=/dev/sda of=sda.mbr.bak bs=512 count=1          # Backup the full MBR
#  dd  if=/dev/zero of=/dev/sda bs=512 count=1             # Delete MBR and partiton table
#  dd  if= sda.mbr.bak of=/dev/sda bs=512 count=1          # Restore the full MBR
#  dd  if= sda.mbr.bak of=/dev/sda bs=446 count=1          # Restore only the boot loader
#  dd  if=sda.mbr.bak of=/dev/sda bs=1 count=64 skip=446    seek=446 # Restore partition table
```

tar

Los archivos creados con tar suelen llevar la extensión tar (no comprimido), tgz (comprimido con gzip) o tbz (comprimido con bzip2). A la hora de crear el archivo NO hay que utilizar rutas absolutas, ya que la restauración sobreescribiría en esas mismas rutas absolutas.

```
#  tar -cf home.tar home/             # archive the whole /home directory (c for create)
#  tar -czf home.tgz home/            # same with zip compression
#  tar -cvjf home.tbz home/            # same with bzip2 compression and verbossely
#  tar  -tzf home.tgz                 # look inside the archive without extracting (list)
#  tar  -xf home.tar                  # extract the archive here (x for extract)
#  tar  -xvzf home.tgz                 # same with zip compression and verbossely
#  tar  -xjf home.tgz                 # same with bzip2 compression
#  tar  -xjf home.tgz home/colin/file.txt         # Restore a single file
```

```
#  tar  c dir/ | gzip  | ssh user@remote 'dd of=dir.tgz' # arch dir/ and store remotely.
```

lpr

```
# lpr unixtoolbox.ps                 # Print on default printer
# export PRINTER=hp4600              # Change the default printer
# lpr -Php4500 #2 unixtoolbox.ps     # Use printer hp4500 and print 2 copies
# lpr -o Duplex=DuplexNoTumble ...   # Print duplex along the long side
# lpr -o PageSize=A4,Duplex=DuplexNoTumble ...
# lpq                                # Check the queue on default printer
# lpq -l -Php4500                    # Queue on printer hp4500 with verbose
# lprm -                             # Remove all users jobs on default printer
# lprm -Php4500 3186                 # Remove job 3186. Find job nbr with lpq
# lpc status                         # List all available printers
# lpc status hp4500                  # Check if printer is online and queue length
```

List installed packages

```
# rpm -qa                                 # List installed packages (RH, SuSE, RPM based)
# dpkg -l                                 # Debian, Ubuntu
```

Add/remove software

Front ends: yast2/yast for SuSE, redhat-config-packages for Red Hat.

```
# rpm -i pkgname.rpm                      # install the package (RH, SuSE, RPM based)
# rpm -e pkgname                          # Remove package
```

Debian

```
# apt-get update                         # First update the package lists
# apt-get install emacs                  # Install the package emacs
# dpkg --remove emacs                    # Remove the package emacs
# dpkg -S file                           # find what package a file belongs to
# aptitude update && aptitude upgrade && aptitude dist-upgrade
```

Gentoo uses emerge as the heart of its "Portage" package management system.

```
# emerge --sync                          # First sync the local portage tree
# emerge -u packagename                  # Install or upgrade a package
# emerge -C packagename                  # Remove the package
# revdep-rebuild                         # Repair dependencies
```

NIVELES DE EJECUCIÓN (*RUNLEVELS*)

Una vez arrancado el sistema, el núcleo inicia el proceso **`init`** (PID 0), que a su vez inicia **`rc`**, quien inicia los *scripts* correspondientes al nivel de ejecución (residentes en **`/etc/init.d`** y enlazados desde **`/etc/rcx.d`**).
El nivel de ejecución por omisión se define en **`/etc/inittab`**. Se puede cambiar el nivel de ejecución con **`init x`**.

A continuación se muestran los diferentes niveles de ejecución y su aplicación en las principales distribuciones. La opción remarcada en negrilla indica el nivel de ejecución por omisión.

Nivel	Genérico	RedHat (Fedora...)	Slackware	Debian (Ubuntu...)
0	Apagar equipo (*Halt*)	Apagar equipo (*Halt*)	Apagar equipo (*Halt*)	Apagar equipo (*Halt*)
1 (S)	Modo de usuario único (*Single-User Mode*)	Modo de usuario único (*Single-User Mode*)	Modo de usuario único (*Single-User Mode*)	Modo de usuario único (*Single-User Mode*)
2	Modo multi-usuario sin red	Personalizable (no se usa)	Personalizable (configurado como n. 3)	**Modo multi-usuario completo**
3	Modo multi-usuario con red	Modo multi-usuario con red	**Modo multi-usuario con red**	Modo multi-usuario completo
4	Personalizable (no se usa)	Personalizable (no se usa)	Modo multi-usuario completo	Personalizable (no se usa)
5	Modo multi-usuario completo (con red y entorno gráfico)	**Modo multi-usuario completo**	Personalizable (configurado como n. 3)	Modo multi-usuario completo
6	Reiniciar equipo (*Reboot*)	Reiniciar equipo (*Reboot*)	Reiniciar equipo (*Reboot*)	Reiniciar equipo (*Reboot*)

Para configurar los scripts de inicio se puede usar **`chkconfig`** en sistemas basados en RedHat o **`update-rc.d`** en sistemas basados en Debian.

```
#  chkconfig  --list                      # List all init scripts
#  chkconfig  --list sshd                 # Report the status of sshd
#  chkconfig  sshd --level 35 on          # Configure sshd for levels 3 and 5
#  chkconfig  sshd off                    # Disable sshd for all runlevels
#  update-rc.d  sshd defaults             # Activate sshd with the default runlevels
#  update-rc.d  sshd start 20 2 3 4 5 . stop 20 0 1 6 . # With explicit arguments
#  update-rc.d  -f sshd remove            # Disable sshd for all runlevels
#  shutdown -h  now (or # poweroff)       # Shutdown and halt the system
```

SISTEMA DE FICHEROS

JERARQUÍA DE DIRECTORIOS (EXTRACTO DE MAN HIER)

/	Este es el directorio raíz. Aquí comienza todo el árbol de directorios.
/bin	Ejecutables que son necesarios en el modo monousuario y para el arranque o reparación del sistema.
/boot	Ficheros estáticos para el cargador de arranque (boot loader).
/dev	Ficheros especiales o de dispositivo, que se refieren a dispositivos físicos. Ver mknod(1).
/dev/fd...	Dispositivos FDD
/dev/hd...	Dispositivos ATAPI
/dev/null	Dispositivo de salida nulo
/dev/random	Dispositivo de entrada que provee aleatoriedad, cuando es posible
/dev/sd...	Dispositivos SCSI o emulados SCSI
/dev/tty...	Dispositivos de consola
/dev/urandom	Dispositivo de entrada que provee pseudo-aleatoriedad, continuamente
/dev/usb...	Dispositivos USB
/dev/zero	Dispositivo de entrada que provee ceros
/etc	Ficheros de configuración locales a la máquina, directamente o en subdirectorios específicos.
/etc/skel	Estructura que se deben copiar al directorio de un nuevo usuario al crearlo.
/home	Directorio donde reside la estructura de directorios "home" de los usuarios (no root).
/lib	Bibliotecas compartidas necesarias para arrancar el sistema y ejecutar las órdenes del sistema.
/media	Estructura de directorios para el montaje automático de dispositivos.
/mnt	Es un punto de montaje para los sistemas de ficheros montados temporalmente.
/proc	Pseudosistema de ficheros con información sobre los procesos en ejecución y el núcleo. Ver proc(5).
/sbin	Como /bin, pero para comandos de uso exclusivo por el superusuario.
/tmp	Contiene ficheros temporales de aplicación que pueden ser borrados sin previo aviso.
/usr	Únicamente debe contener datos compartidos de sólo lectura.
/usr/include	Ficheros cabecera para el compilador C.
/usr/lib	Bibliotecas objeto. Algunos programas pueden tener subdirectorios específicos.
/usr/local	Contiene los programas que son locales a la instalación.
/usr/local/bin	Contiene los binarios de los programas locales de la instalación.
/usr/man	Contiene las páginas del manual, en sus subdirectorios.
/usr/sbin	Contiene binarios de los programas de administración que no son esenciales.
/usr/src	Ficheros fuente para diferentes partes del sistema.
/usr/src/linux	Contiene las fuentes del núcleo del sistema operativo propiamente dicho.
/var	Contiene ficheros cambian de tamaño habitualmente, como los ficheros de colas y de registro (log).
/var/lock	Contiene los ficheros de bloqueo.
/var/log	Contiene los ficheros de registro (*logs*).
/var/spool	Colas de ficheros para diversos programas.

MONTAJE DE SISTEMAS DE FICHEROS

Los puntos de montaje se definen en **`/etc/fstab`**:

```
/dev/cdrom  /media/cdrom  auto noauto,user,ro,procuid,nosuid,nodev,exec 0 0
```

```
mount | cut -d ' ' -f1,3,5-6 | column -t     # Muestra FS montados | se 'embellece' la salida
umount /home/                                # Desmonta /home
mount  -t auto /dev/cdrom /media/cdrom       # Monta un cdrom
mount  /dev/hdc -t iso9660 -r /cdrom         # Monta un cdrom ATAPI
mount  /dev/sdc -t iso9660 -r /cdrom         # Monta un cdrom SCSI
mount /dev/sda1 /media/windowsC -o utf8      # Monta una partición Windows
mount -o remount,ro /                        # Remonta la raiz en modo solo lectura
smbmount //winserv/guimi mount/guimi -o username=guimi,workgroup=GUIMINET
mount -t cifs //winserv/guimi mount/guimi -o username=guimi,workgroup=GUIMINET
mount -t iso9660 -o loop file.iso /mnt       # Monta una imagen de CD
```

Encontrar ficheros abiertos

Por ejemplo para cerrar procesos y poder desmontar

```
fuser -m /media/USBDrive      # Lista los procesos con ficheros abiertos en /media/USBDrive
lsof -p 3324                  # Lista ficheros abiertos por el proceso 3324
```

```
lsof /media/USBDrive          # Lista procesos con ficheros abiertos en /home
lsof /var/log/Xorg.0.log      # Lista procesos con /var/log/Xorg.0.log abierto
```

Rendimiento de discos

```
time dd if=/dev/hda1 of=/dev/null bs=1024k count=1000
time dd if=/dev/zero bs=1024k count=1000 of=/home/1Gb.file
hdparm -tT /dev/hda
```

RED

GESTIÓN

Interfaces

```
#  ip link show              #  Display all interfaces on Linux (similar to ifconfig)
#  ip link set eth0 up       #  Bring device up (or down). Same as "ifconfig eth0 up"
#  ip addr show              #  Display all IP addresses on Linux (similar to ifconfig)
#  ip neigh show             #  Similar to arp -a
#  ifconfig eth0
```

```
# arp -a                    # Check the router (or host) ARP entry (all OS)
# ping guimi.net
# traceroute guimi.net
# netstat -s                # System-wide statistics for each network protocol
# arping 192.168.16.254      # Ping on ethernet layer
```

```
# ifconfig eth0 192.168.50.254 netmask 255.255.255.0               # First IP
# ifconfig eth0:0 192.168.51.254 netmask 255.255.255.0             # Second IP
# ip addr add 192.168.50.254/24 dev eth0                           # Equivalent ip commands
# ip addr add 192.168.51.254/24 dev eth0 label eth0:1
# ifconfig eth0      down
# ifconfig eth0      hw ether 00:01:02:03:04:05         # Linux
```

Encaminamiento

```
# route -n                   # Linux or use "ip route"
# netstat -rn                # Linux, BSD and UNIX
#  route add -net 192.168.20.0 netmask 255.255.255.0 gw 192.168.16.254
#  ip route add 192.168.20.0/24 via 192.168.16.254        # same as above with ip route
#  route add -net 192.168.20.0 netmask 255.255.255.0 dev eth0
#  route add default gw 192.168.51.254
#  ip route add default via 192.168.51.254 dev eth0       # same as above with ip route
#  route delete -net 192.168.20.0 netmask 255.255.255.0
```

Puertos en uso

```
# netstat -an | grep LISTEN
# lsof -i                          # Linux list all Internet connections
# socklist                         # Linux display list of open sockets
# netstat -anp --udp --tcp | grep LISTEN                # Linux
# netstat -tup                     # List active connections to/from system (Linux)
# netstat -tupl                    # List listening ports from system (Linux)
```

Cortafuegos y Pasarelas

```
# iptables -L -n -v                     # For status
# echo 1 > /proc/sys/net/ipv4/ip_forward # IP forward 0=off, 1=on
# echo 1 > /proc/sys/net/ipv4/icmp_echo_ignore_broadcasts
# iptables -L -t nat                    # Check NAT status
```

DHCP

Como cliente de DHCP algunas distribuciones, como SuSE, utilizan **`dhcpcd`**:

```
# dhcpcd -n eth0                 # Trigger a renew (does not always work)
# dhcpcd -k eth0                 # release and shutdown
```

La información recibida se guarda en: **`/var/lib/dhcpcd/dhcpcd-eth0.info`**.

Otras distribuciones, como Debian, utilizan **`dhclient`**:

```
# dhclient eth0                   # Use /etc/dhclient.conf
```

La información recibida se guarda en: **`/var/lib/dhcp3/dhclient.eth0.leases`**.

RESOLUCIÓN DE NOMBRES

/etc/hosts

El fichero hosts permite resolver direcciones de forma local. Tiene preferencia sobre el servicio DNS, aunque es configurable en **`/etc/nsswitch.conf`** y **`/etc/host.conf`**.

```
78.31.70.238       sleepyowl.net     sleepyowl
```

DNS

La configuración DNS reside en **`/etc/resolv.conf`** y es válida para todas las interfaces.

```
nameserver 78.31.70.238
search sleepyowl.net intern.lab
domain sleepyowl.net
```

```
# hostname -d                           # Same as dnsdomainname
```

```
# host -t MX guimi.net                  # Get the mail MX entry
# host -t NS -T guimi.net               # Get the NS record over a TCP connection
# host -a guimi.net                     # Get everything
# dig  guimi.net
# dig -x 78.31.70.238
# host 78.31.70.238
# nslookup 78.31.70.238
```

SSH

CONFIGURACIÓN

ssh (y sus familiares **scp**, **sftp**) permiten realizar conexiones seguras entre equipos, autenticando las conexiones mediante pares de usuario-contraseña, kerberos o pares de claves pública-privada.
Para mayor seguridad cambiamos el puerto del servidor ssh, reducimos el tiempo de login y no permitimos conexiones de root:

```
# vi /etc/ssh/sshd_config
Port xxxx
LoginGraceTime 45
PermitRootLogin no
```

Solo permitimos acceder por ssh a los usuarios indicados en el fichero /etc/loginusers:

```
# vi /etc/pam.d/ssh
#auth       required     pam_env.so # [1]
auth       required     pam_listfile.so sense=allow onerr=fail item=user file=/etc/loginusers
```

Como usuario generamos las claves dsa y asignamos una passphrase

```
$ ssh-keygen -t dsa
```

Para cambiar (o des/asignar) una passphrase (se recomienda cambiarla regularmente):

```
$ ssh-keygen -p -f .ssh/id_dsa
```

Autorizamos la clave publica:

```
$ cat .ssh/id_dsa.pub > .ssh/authorized_keys
```

La clave privada (id_dsa) se tiene que copiar en los clientes de ssh a través de algún medio seguro.
A partir de este momento es posible conectar desde otra maquina utilizando login y contraseña como siempre:

```
$ ssh usuario@maquina -p xxxx
```

o usando el fichero de clave privada y conociendo la passphrase:

```
$ ssh usuario@maquina -p xxxx -i id_dsa_copiada
```

Para permitir solo el uso de certificados reconfiguramos sshd desautorizando las autenticaciones por contraseña:

```
# vi /etc/ssh/sshd_config
PasswordAuthentication no
```

Reiniciamos el servicio:

```
# /etc/init.d/ssh restart
```

CONFIGURACIÓN DE PuTTY

Para usar la clave privada en PuTTY (Windows):

1. Copiamos la clave al equipo
2. Usamos PuTTYGen para importar la clave y guardarla como .ppk
   - Conversions->Import key
   - Introducir la passphrase
   - Pulsar "Save private key" y guardarla con extensión .ppk
3. Al hacer la conexión, en el menú SSH->Auth seleccionamos nuestro fichero .ppk
4. Si al conectar muestra el error "Key is of wrong type (PuTTY SSH2 private key)", seleccionar en PuTTY Connections->SSH el protocolo "2".

USO

La primera vez que se realiza una conexión a un servidor ssh, el cliente nos avisa de que no puede autenticar al servidor y nos muestra la firma de su clave pública ("*fingerprint*"). Una vez aceptada la conexión, si cambia la clave del servidor, el cliente no nos dejará conectar (deberemos borrar manualmente la clave guardada).

Para evitar un ataque tipo "*man-in-the-middle*" el administrador del servidor puede obtener y enviar la firma de su clave:

```
 # ssh-keygen -l -f /etc/ssh/ssh_host_dsa_key.pub              # For DSA key (default)
```

También se puede usar de manera similar los comandos **scp** y **sftp** (interactivo).

```
scp file.txt host-two:/tmp
scp joe@host-two:/www/*.html /www/tmp
scp -r joe@host-two:/www /www/tmp
```

GESTIÓN DE TÚNELES

La gestión de túneles de **ssh** permite redireccionar un puerto local o remoto sobre la conexión de túnel de SSH, asegurando la comunicación. Es posible redirigir más de un puerto. De forma predefinida SSH redirige la comunicación con el servidor X.

```
ssh -X user@gate          # To force X forwarding
```

Por ejemplo, supongamos que desde la máquina clie1 queremos utilizar un programa que accede a una BB.DD y que el servidor serv1 sirve de forma insegura la BB.DD. en el puerto 3306 (MySQL) y dispone de un servidor SSH.
Desde el cliente clie1 redirigimos el puerto remoto 3306 al puerto local 13306. Así en la máquina cliente configuramos al programa para que conecte a la BB.DD. del puerto local 13306 (localhost:13306). De este forma, de manera totalmente transparente para la aplicación, todas las comunicaciones entre clie1 y serv1 quedarán protegidas por el túnel SSH. La sintaxis es:

```
ssh -L localport:desthost_name:destport usuario@desthost
```

Hay que tener en cuenta que quien resuelve "desthost_name" es "desthost" por lo que son servirían:

```
clie1$ ssh -L 13306:serv1:3306 usuario@serv1
clie1$ ssh -L 13306:localhost:3306 usuario@serv1 # Quien resuelve "localhost" es serv1
```

En este ejemplo tunelamos una conexión http (puerto 80) y CVS (puerto 2401):

```
ssh -L 2401:localhost:2401 -L 8080:localhost:80 user@gate
```

Tambien se puede hacer lo contrario "Reverse-Forwarding" para enviar un puerto local a una máquina remota:

```
ssh -R destport:desthost_name:localport user@desthost
```

Que "desthost_name" lo calcule "desthost" nos permite redirigir puertos a otras máquinas, configurando una conexión tipo VPN de acceso remoto, en la que el cliente conecta con la red tras la pasarela (servidor SSH):
Por ejemplo, se puede redirigir a una máquina "smbserver" destrás de una pasarela (*gateway*) los puertos de remote desktop (139) y de compartición smb (3389):

```
ssh -L 139:smbserver:139 -L 3388:smbserver:3389 user@gate
```

Solo las comunicaciones entre el cliente y la pasarela irán por tunel SSH, no las conexiones entre la pasarela y smbserver.

VPN SOBRE SSH

Desde la versión 4.3 OpenSSH permite utilizar la interfaz virtual de red tun/tap para cifrar un túnel, de manera similar a como hacen otras soluciones VPN sobre TLS. TAP simula un dispositivo Ethernet y opera en el nivel 2, mientras que TUN (*TUNnel*) simula una capa de red de nivel 3. TAP se usa para crear el puente de red y TUN para encaminar.
Algunas ventajas de usar SSH son que no hace falta usar ningún otro software, que el túnel utiliza la autenticación de SSH, permitiendo por ejemplo usar claves públicas y privadas y que permite enviar cualquier protocolo de nivel 3 y 4, como ICMP o TCP/UDP.
A cambio el encapsulamiento se realiza sobre TCP, obteniéndose peor rendimiento que con los túneles IPSec, además de depender de conexiones TCP.
Hace falta las siguientes opciones en ssd_config:

```
PermitRootLogin yes          # Para determinadas configuraciones puede ser necesario o no
PermitTunnel yes
```

Túnel entre equipos (P2P)

No es estrictamente una VPN. Vamos a conectar dos equipos hclient y hserver. La conexión se inicia desde hclient hacia hserver y la realiza root.
El túnel termina en las direcciones 10.0.1.1 (hserver) y 10.0.1.2 (hclient) y creamos un dispositivo tun5. El procedimiento es muy sencillo:

1. Conectamos por SSH usando la opción de túnel (-w)
2. Configuramos las IPs del túnel.

Conectamos desde el cliente y ejecutamos comandos en el servidor para configurarlo:

```
cli>#  ssh -w5:5 root@hserver
srv>#  ifconfig tun5 10.0.1.1 netmask 255.255.255.252         # Executed on the server shell
```

Configuramos el cliente:

```
cli>#  ifconfig tun5 10.0.1.2 netmask 255.255.255.252
```

Ahora los dos equipos están conectados y pueden comunicarse de manera transparente con cualquier protocolo de capa 3 o 4 utilizando las direcciones IP del túnel.

Túnel entre redes

Para hacer un túnel entre dos redes, por ejemplo netA (192.168.51.0/24) y netB (192.168.16.0/24), utilizaremos dos pasarelas SSH. El procedimiento es similar al anterior, pero hemos de añadir encaminamiento. Además debe activarse NAT en la interfaz privada de las pasarelas VPN SSH solo si no son las pasarelas predefinidas de las redes.

192.168.51.0/24 (netA) | gateA <-> gateB | 192.168.16.0/24 (netB)

1. Conectamos por SSH usando la opción de túnel (-w)
2. Configuramos las IPs del túnel.
3. Añadimos encaminamiento para las dos redes.
4. Si es necesario, activamos NAT en la interfaz privada de la pasarela.

Empezamos la comunicación desde la red netA. Conectamos desde gateA y ejecutamos comandos en gateB:

```
gateA>#  ssh -w5:5 root@gateB
gateB>#  ifconfig tun5 10.0.1.1 netmask 255.255.255.252 # Executed on the gateB shell
gateB>#  route add -net 192.168.51.0 netmask 255.255.255.0 dev tun5
gateB>#  echo 1 > /proc/sys/net/ipv4/ip_forward           # Only needed if not default gw
gateB>#  iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE
```

Configuramos gateA:

```
gateA>#  ifconfig tun5 10.0.1.2 netmask 255.255.255.252
gateA>#  route add -net 192.168.16.0 netmask 255.255.255.0 dev tun5
gateA>#  echo 1 > /proc/sys/net/ipv4/ip_forward
gateA>#  iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE
```

Las dos redes están ahora conectadas de manera transparente a través del túnel VPN SSH. Si las pasarelas VPN SSH no son las pasarelas de las redes, los clientes no sabrán a qué dirección enviar las respuestas, por eso activamos NAT.

DISK QUOTA

The quota tools package usually needs to be installed, it contains the command line tools. Activate the user quota in the fstab and remount the partition. If the partition is busy, either all locked files must be closed, or the system must be rebooted. Add usrquota to the fstab mount options, for example:

```
/dev/sda2       /home     reiserfs       rw,acl,user_xattr,usrquota 1 1
# mount -o remount /home
```

Initialize the quota.user file with quotacheck.

```
# quotacheck -vum /home
# chmod 644 /home/aquota.user              # To let the users check their own quota
```

Activate the quota either with the provided script (e.g. /etc/init.d/quotad on SuSE) or with quotaon:

```
quotaon -vu /home
```

Check that the quota is active with:

```
quota -v
```

Assign quota limits

The quotas are not limited per default (set to 0). The limits are set with edquota for single users. A quota can be also duplicated to many users. The file structure is different between the quota implementations, but the principle is the same: the values of blocks and inodes can be limited.
Only change the values of soft and hard. If not specified, the blocks are 1k. The grace period is set with edquota -t. For example:

```
# edquota -u colin
```

Linux

```
Disk quotas for user colin (uid 1007):
  Filesystem           blocks          soft        hard     inodes     soft     hard
  /dev/sda8               108         1000        2000          1        0        0
```

Users can check their quota by simply typing quota (the file quota.user must be readable). Root can check all quotas.

```
# quota -u colin                          # Check quota for a user
# repquota /home                          # Full report for the partition for all users
```

```
# setquota
```

RAID 1

Supongamos que tenemos en **/dev/sda** nuestro sistema básico instalado y queremos utilizar **/dev/sdb** como disco espejo.
Instalamos las herramientas de RAID:

```
# aptitude install mdadm
```

Desmontamos todas las particiones excepto "/" y swap. Utilizamos fdisk para cambiar los tipos de las particiones a "fd" (raid autodetect). **No hay que cambiar la partición swap**.
Veremos un error al escribir la nueva tabla por estar el sistema en uso, pero no pasa nada. Verificamos:

```
# fdisk -l /dev/sda
Disk /dev/sda: 160.0 GB, 160041885696 bytes
255 heads, 63 sectors/track, 19457 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes

   Device Boot      Start         End      Blocks   Id  System
/dev/sda1   *           1        3647    29294496   fd  Linux raid autodetect
/dev/sda2            3648        7294    29294527+  fd  Linux raid autodetect
[...]
```

Usamos sfdisk para copiar la tabla de particiones en /dev/sdb:

```
# sfdisk -d /dev/sda | sfdisk /dev/sdb
```

Creamos los volúmenes RAID indicando que se componen de dos dispositivos. El primero está perdido (no queremos aún que utilice las particiones donde hemos instalado) y el segundo está en /dev/sdb:

```
# mdadm --create /dev/md0 --level 1 --raid-devices=2 missing /dev/sdb1
# mdadm --create /dev/md1 --level 1 --raid-devices=2 missing /dev/sdb2
```

Creamos sistemas de fichero en los volúmenes RAID:

```
# mkfs.ext3 /dev/md0
# mkfs.ext3 /dev/md1
```

Montamos el volumen que contendrá el sistema y copiamos la raiz:

```
# mount /dev/md0 /mnt
# cp -dpRx / /mnt
```

Montamos y copiamos el resto de volúmenes:

```
# mount /dev/md1 /mnt/home
# cp -dpRx /home /mnt/home
```

Editamos /etc/fstab:

```
/dev/md0       /              ext3    defaults,errors=remount-ro 0       1
/dev/md1       /home          ext3    nodev,nosuid,noexec,usrquota,grpquota         0       2
/dev/sda3       none            swap    sw,pri=1        0       0
/dev/sdb3       none            swap    sw,pri=1        0       0
```

Como se ve hemos preparado md1 para utilizar cuotas y se han configurado por seguridad para que en esos sistemas de ficheros no se utilicen dispositivos (nodev), ficheros con setuid (nosuid) ni ficheros ejecutables (noexec).

Editamos /boot/grub/menu.lst. Opcionalmente podemos activar *framebuffer*:

```
(...)
title           Debian GNU/Linux, kernel 2.6.18-3-686
root            (hd0,0)
kernel          /boot/vmlinuz-2.6.18-3-686 root=/dev/md0 md=0,/dev/sda1,/dev/sdb1 ro vga=791
initrd          /boot/initrd.img-2.6.18-3-686
savedefault

title           Debian GNU/Linux, kernel 2.6.18-3-686 (RAID recovery disc-A)
root            (hd0,0)
kernel          /boot/vmlinuz-2.6.18-3-686 root=/dev/md0 md=0,/dev/sda1 ro single
initrd          /boot/initrd.img-2.6.18-3-686
savedefault

title           Debian GNU/Linux, kernel 2.6.18-3-686 (RAID recovery disc-B)
root            (hd1,0)
kernel          /boot/vmlinuz-2.6.18-3-686 root=/dev/md0 md=0,/dev/sdb1 ro single
initrd          /boot/initrd.img-2.6.18-3-686
```

```
savedefault
(...)
```

Copiamos los ficheros editados al disco espejo:

```
# cp -dp /etc/fstab /mnt/etc/fstab
# cp -dp /boot/grub/menu.lst /mnt/boot/grub
```

Instalamos **grub** en el segundo disco, para que podamos arrancar si falla el primero:

```
# grub-install /dev/sda
# grub
grub> device (hd0) /dev/sdb
grub> root (hd0,0)
grub> setup (hd0)
grub> quit
```

Y por fin reiniciamos el sistema para utilizar ya los volúmenes RAID y verificar que todo va bien. El sistema debería rearrancar correctamente. Algunas verificaciones que podemos hacer:

```
# mount
# df
# mdadm -QD /dev/md0
# mdadm -QD /dev/md1
# cat /proc/mdstat
```

Añadimos las particiones de /dev/sda (que no estamos usando) a los volúmenes RAID:

```
# mdadm --add /dev/md0 /dev/sda1
# mdadm --add /dev/md1 /dev/sda2
```

Cada vez que añadimos una partición a un volumen RAID debe sincronizarla. Si se intenta añadir una partición antes de que termine de sincronizar la anterior el sistema nos dará un aviso y encolará la tarea.
No reiniciar hasta que las particiones estén sincronizadas.
Podemos verlo en /proc/mdstat. Cuando indica [UU] es que está correcto.

```
# cat /proc/mdstat
Personalities : [linear] [multipath] [raid0] [raid1] [raid6] [raid5] [raid4] [raid10]
md0 : active raid1 sda1[0] sdb1[1]
      29294400 blocks [2/2] [UU]

md1 : active raid1 sda2[0] sdb2[1]
      29294400 blocks [2/2] [UU]

unused devices: <none>
```

Si se desea quitar una partición de un volumen:

```
# mdadm --fail /dev/md2 /dev/sda4
# mdadm --remove /dev/md2 /dev/sda4
```

No se pueden quitar todas las particiones de un volumen. Si lo que se desea es eliminar el volumen:

```
# mdadm --stop /dev/md2
```

Nada de esto afecta al estado de las particiones en sí mismas.

FICHEROS DE CONFIGURACIÓN

```
$ cat /etc/apt/sources.list
deb http://ftp.fr.debian.org/debian/ etch main contrib non-free
deb-src http://ftp.fr.debian.org/debian/ etch main contrib

deb http://security.debian.org/ etch/updates main contrib
deb-src http://security.debian.org/ etch/updates main contrib

deb http://www.debian-multimedia.org/ etch main
deb-src http://www.debian-multimedia.org/ etch main
```

```
$ cat /etc/bash.bashrc
# System-wide .bashrc file for interactive bash(1) shells.
[...]

# If not running interactively, don't do anything
[ -z "$PS1" ] && return

# set a fancy prompt (non-color, overwrite the one in /etc/profile)
PS1='${debian_chroot:+($debian_chroot)}\u@\h:\w\$ '
[...]

if [ -d ~/bin ]; then
        export PATH=$PATH:~/bin
fi
```

```
$ cat /etc/crontab
# /etc/crontab: system-wide crontab
[...]
SHELL=/bin/sh
PATH=/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin

# m h dom mon dow user  command
17 *    * * *   root    cd / && run-parts --report /etc/cron.hourly
25 6    * * *   root    test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.daily )
47 6    * * 7   root    test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.weekly )
52 6    1 * *   root    test -x /usr/sbin/anacron || ( cd / && run-parts --report
/etc/cron.monthly )
```

```
$ cat /etc/dhcpd.conf
option domain-name "Mi_Dominio";
option domain-name-servers Servidor.Mi_Dominio;
option subnet-mask 255.255.255.224;
default-lease-time 600;
max-lease-time 7200;

subnet 280.280.280.0 netmask 255.255.255.0 {
    range 280.280.280.100 280.280.280.250;
    option subnet-mask 255.255.255.0;
    option broadcast-address 280.280.280.255;
    option routers 280.280.280.1;
    option domain-name-servers 280.280.280.333;
}
```

```
$ cat /etc/exports
# /etc/exports: the access control list for filesystems which may be exported
#               to NFS clients.  See exports(5).
```

```
$ cat /etc/fstab
# /etc/fstab: static file system information.
#
# <file system> <mount point>   <type>  <options>       <dump>  <pass>
proc            /proc           proc    defaults        0       0
/dev/sda2       /               ext3    defaults,errors=remount-ro 0       1
/dev/sda5       /media/DATOS    auto    defaults,errors=remount-ro 0       0
/dev/sda6       /home           ext3    defaults        0       2
/dev/sda7       none            swap    sw              0       0
/dev/hda        /media/cdrom0   udf,iso9660 user,noauto     0       0
/dev/fd0        /media/floppy0  auto    rw,user,noauto  0       0

/dev/sda1      /media/windowsC  ntfs    user,noauto,ro,dmask=000,fmask=111,nls=utf8,gid=100 0 0
```

```
$ cat /etc/group
root:x:0:
daemon:x:1:
bin:x:2:
sys:x:3:
adm:x:4:guimi
tty:x:5:
disk:x:6:guimi
[...]
```

```
$ cat /etc/hosts
127.0.0.1 localhost
158.42.xxx.x xxxxxxxxx.upv.es
```

```
$ cat /etc/hostname
xxxxxxxx.upv.es
```

```
$ cat /etc/hosts.allow
# /etc/hosts.allow: list of hosts that are allowed to access the system.
[...]
ALL: .foobar.edu EXCEPT terminalserver.foobar.edu
```

```
$ cat /etc/hosts.deny
# /etc/hosts.deny: list of hosts that are _not_ allowed to access the system.
[...]
ALL: PARANOID
```

```
# touch /etc/hosts.equiv /etc/ssh/shosts.equiv && chmod 000 /etc/hosts.equiv /etc/ssh/shosts.equiv
```

```
$ cat /etc/inittab
# /etc/inittab: init(8) configuration.
[...]
# The default runlevel.
id:3:initdefault:

# Boot-time system configuration/initialization script.
# This is run first except when booting in emergency (-b) mode.
si::sysinit:/etc/init.d/rcS

# What to do in single-user mode.
~~:S:wait:/sbin/sulogin
[...]

l0:0:wait:/etc/init.d/rc 0
l1:1:wait:/etc/init.d/rc 1
l2:2:wait:/etc/init.d/rc 2
[...]

# What to do when CTRL-ALT-DEL is pressed.
ca:12345:ctrlaltdel:/sbin/shutdown -t1 -a -r now
[...]

1:2345:respawn:/sbin/getty 38400 tty1
2:23:respawn:/sbin/getty 38400 tty2
3:23:respawn:/sbin/getty 38400 tty3
[...]
```

```
$ cat /etc/modules
# /etc/modules: kernel modules to load at boot time.
loop
```

```
$ cat /etc/network/interfaces
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).
# Si se usa NetworkManager estas configuraciones pueden no funcionar correctamente.

# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
auto eth0
#iface eth0 inet dhcp
iface eth0 inet static
    address 158.42.xxxxx
    network 158.42.xxxxx
    netmask 255.255.255.0
    broadcast 158.42.xxx.255
    gateway 158.42.xxxxxxx
```

```
$ cat /etc/nsswitch.conf
# /etc/nsswitch.conf

passwd:         files ldap
group:          files ldap
```

```
$ cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/bin/sh
bin:x:2:2:bin:/bin:/bin/sh
sys:x:3:3:sys:/dev:/bin/sh
[...]
guimi:x:1000:1000:guimi,,,:/home/guimi:/bin/bash
```

```
$ cat /etc/profile
# /etc/profile: system-wide .profile file for the Bourne shell (sh(1))
# and Bourne compatible shells (bash(1), ksh(1), ash(1), ...).
```

```
$ cat /etc/rc.local
#!/bin/sh -e
# This script is executed at the end of each multiuser runlevel.
```

```
$ cat /etc/resolv.conf
# generated by NetworkManager, do not edit!
domain upv.es
nameserver 158.42.xxxxxxx
nameserver 158.42.xxxxxxx
```

```
$ cat /etc/services
# Network services, Internet style
# Updated from http://www.iana.org/assignments/port-numbers and other
# sources like http://www.freebsd.org/cgi/cvsweb.cgi/src/etc/services.
tcpmux      1/tcp               # TCP port service multiplexer
echo        7/tcp
echo        7/udp
discard     9/tcp       sink null
discard     9/udp       sink null
[...]
```

```
# cat /etc/shadow
root:xxxxxxxxxxxxxxxxxxxxx3s26LwGhabI30:13777:0:99999:7:::
daemon:*:13650:0:99999:7:::
[...]
guimi:xxxxxxxxxxxxxxxxxxxxxxxxxxh1vOBgX0:13838:0:99999:7:::
```

```
# cat /etc/sudoers
# /etc/sudoers
# This file MUST be edited with the 'visudo' command as root.
[...+
```

```
$ cat /etc/timezone
Europe/Madrid
```

```
$ cat /etc/syslog.conf
#  /etc/syslog.conf     Configuration file for syslogd.
[...]

auth,authpriv.*                 /var/log/auth.log
*.*;auth,authpriv.none          -/var/log/syslog
[...]

daemon.*;mail.*;\
        news.crit;news.err;news.notice;\
        *.=debug;*.=info;\
        *.=notice;*.=warn       /dev/tty12
```

Uso de pam

Uso de xinetd

update-inetd --disable chargen
update-inetd --disable ident
update-rc.d -f cups remove
update-rc.d -f gdm remove

nmap localhost -p 1-5000,8110,10024-10026

uso lsattr chattr

ACLs

ACL de archivos en Linux
Posteado en Debian, Seguridad por situ el 12 de February de 2009

Muchas veces tenemos que dar permisos sobre nuestros archivos a ciertos usuarios del sistema, pero para que esto nos funcione debemos ingresarlos al mismo grupo de trabajo que nosotros, pero con esto estamos iniciando un gran fallo de seguridad ya que realizando el procedimiento mencionado este usuario va a poder ver minimamente nuestros archivos. Para mitigar dicho problema, vamos a utilizar las herramientas “acl”.

1. Comprobamos el soporte de ACL en nuestro kernel.
# cat /boot/config-2.6.18-6-686-bigmem | grep _ACL
CONFIG_EXT2_FS_POSIX_ACL=y
CONFIG_EXT3_FS_POSIX_ACL=y
CONFIG_REISERFS_FS_POSIX_ACL=y
CONFIG_JFS_POSIX_ACL=y
CONFIG_FS_POSIX_ACL=y
CONFIG_XFS_POSIX_ACL=y
CONFIG_JFFS2_FS_POSIX_ACL=y
CONFIG_NFS_V3_ACL=y
CONFIG_NFSD_V2_ACL=y
CONFIG_NFSD_V3_ACL=y
CONFIG_NFS_ACL_SUPPORT=m

2. Instalamos la aplicacion ya que vemos que tenemos soporte de para utlizarlas.

apt-get install acl
Luego debemos remontar nuestro sistema de archivos con soporte acl, para que se pueda trabajar con ellas.
mount -o remount,acl /
3. Comprobamos los permisos del archivo permisos.txt
ls -l permisos.txt
-rwx—— 1 root root 5 2009-02-11 22:36 permisos.txt
getfacl permisos.txt
file: permisos.txt
owner: root
group: root
user::rwx
group::—
other::—

4. Tratamos de leer el contenido del archivo usando el usuario situ.
cat permisos.txt
cat: permisos.txt: Permission denied
Como podemos ver no tenemos permisos para leer el contenido del archivo en cuestion.

4. Damos autorizacion al usuario situ a ver el contenido
setfacl -m u:situ:r permisos.txt
Verificamos los permisos
getfacl permisos.txt
file: permisos.txt
owner: root
group: root
user::rwx
user:situ:r–
group::—
mask::r–
other::—

Ejecutamos nuevamente el cat y ahora podemos ver el contenido.
cat permisos.txt
hola
Para mas info: http://www.debianhelp.co.uk/acl.htm

ENTORNOS GRÁFICOS

CDE (Common Desktop Enviroment)
fvwm
xfce
GNOME (GNU)
KDE (KDE Desktop)