

# Comandos GNU/Linux e SHELL BASH (/bin/bash)

terminator

```
user@computer: ~ 82x24
top - 23:30:36 up 3:26, 0 users, load average: 1,12, 1,18, 1,23
Tasks: 277 total, 2 running, 275 sleeping, 0 stopped, 0 zombie
%Cpu0 : 0,0 us, 0,0 sy, 0,0 ni, 100,0 id, 0,0 wa, 0,0 hi, 0,0 si, 0,0 st
%Cpu1 : 1,7 us, 0,3 sy, 0,0 ni, 91,0 id, 7,0 wa, 0,0 hi, 0,0 si, 0,0 st
%Cpu2 : 99,3 us, 0,7 sy, 0,0 ni, 0,0 id, 0,0 wa, 0,0 hi, 0,0 si, 0,0 st
%Cpu3 : 0,7 us, 0,3 sy, 0,0 ni, 99,0 id, 0,0 wa, 0,0 hi, 0,0 si, 0,0 st
KiB Mem: 3890584 total, 2629160 used, 1261424 free, 333924 buffers
KiB Swap: 2559996 total, 0 used, 2559996 free. 984516 cached Mem

  PID USER      PR  NI   VIRT   RES   SHR  S  %CPU  %MEM     TIME+ COMMAND
 3118 user        20   0   5764   1328  1240 R 100,0  0,0   74:14.74 yes
1365 root         20   0  447536  8768  7668 S   0,7  0,2   0:33.04 gvfsd-trash
1974 user        20   0 1981900 227836 55620 S   0,7  5,9   2:18.69 gnome-shell
 411 root        -51   0     0     0     0 S   0,3  0,0   0:06.97 irq/30-iwlwi+
1294 root         20   0  533228  34392 23320 S   0,3  0,9   0:00.79 nm-applet
1705 root         20   0  297288  49880 36332 S   0,3  1,3   1:19.41 Xorg
2010 user        20   0  729972  48340 28644 S   0,3  1,2   0:04.91 nautilus
2087 user        20   0  447536  8480   7372 S   0,3  0,2   0:32.94 gvfsd-trash
5384 root         20   0     0     0     0 S   0,3  0,0   0:00.15 kworker/u8:0
5534 user        20   0  25800   3112  2496 R   0,3  0,1   0:01.30 top
   1 root         20   0  177548  5772  3092 S   0,0  0,1   0:01.67 systemd
   2 root         20   0     0     0     0 S   0,0  0,0   0:00.00 kthreadd
   3 root         20   0     0     0     0 S   0,0  0,0   0:00.92 ksoftirqd/0
   5 root         20  -20     0     0     0 S   0,0  0,0   0:00.00 kworker/0:0H

root@computer: ~ 8
user@computer:~$ id
uid=1000(user) gid=1001(user) grupos=1001(user),14
user@computer:~$ su -
Contraseña:
root@computer:~$ id
uid=0(root) gid=0(root) grupos=0(root)
root@computer:~$

user@computer: ~ 8
user@computer:~$ ping -c2 www.google.es
PING www.google.es (178.60.128.20) 56(84) bytes of
64 bytes from cache.google.com (178.60.128.20): ic
64 bytes from cache.google.com (178.60.128.20): ic

--- www.google.es ping statistics ---
2 packets transmitted, 2 received, 0% packet loss,
rtt min/avg/max/mdev = 50.088/50.432/50.777/0.411
user@computer:~$
```

gnome-terminal

```
user@computer: ~
Archivo Editar Ver Buscar Terminal Ayuda
Every 1,0s: ip addr show eth0 Sat Jan 16 00:04:22 2016
2: eth0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc pfifo_fast state DOWN group default qlen 1000
link/ether 08:9e:01:11:9d:cc brd ff:ff:ff:ff:ff:ff

top - 00:04:21 up 4:00, 0 users, load average: 1,39, 1,25, 1,24
Tasks: 278 total, 2 running, 276 sleeping, 0 stopped, 0 zombie
%Cpu(s): 28,4 us, 3,3 sy, 0,0 ni, 65,9 id, 2,3 wa, 0,0 hi, 0,0 si, 0,0 st
KiB Mem: 3890584 total, 2637656 used, 1252928 free, 339636 buffers
KiB Swap: 2559996 total, 0 used, 2559996 free. 965112 cached Mem

  PID USER      PR  NI   VIRT   RES   SHR  S  %CPU  %MEM     TIME+ COMMAND
 3118 user        20   0   5764   1328  1240 R 100,0  0,0 107:59.28 yes
1974 user        20   0 1989872 237708 56120 S  14,3  6,1   3:09.74 gnome-shell
1705 root         20   0  300376  48400 33788 S   8,0  1,2   1:35.56 Xorg
7323 user        20   0  346828  30192 21864 S   1,7  0,8   0:00.42 gnome-scre+
1365 root         20   0  447536  8768  7668 S   1,0  0,2   0:47.33 gvfsd-trash
2087 user        20   0  447536  8488  7372 S   1,0  0,2   0:47.21 gvfsd-trash
 411 root        -51   0     0     0     0 S   0,3  0,0   0:08.04 irq/30-iwl+
 801 message+    20   0  43964  4968  2992 S   0,3  0,1   0:01.91 dbus-daemon
 840 redis       20   0  38200  3092  2272 S   0,3  0,1   0:07.21 redis-serv+
1881 user        20   0  42964  3340  2064 S   0,3  0,1   0:01.44 dbus-daemon
   1 root         20   0  177548  5772  3092 S   0,0  0,1   0:01.70 systemd
   2 root         20   0     0     0     0 S   0,0  0,0   0:00.00 kthreadd
   3 root         20   0     0     0     0 S   0,0  0,0   0:01.08 ksoftirqd/0
   5 root         20  -20     0     0     0 S   0,0  0,0   0:00.00 kworker/0:+
   7 root         20   0     0     0     0 S   0,0  0,0   0:06.90 rcu_sched
   8 root         20   0     0     0     0 S   0,0  0,0   0:00.00 rcu_bh
   9 root         rt   0     0     0     0 S   0,0  0,0   0:00.01 migration/0
```

Ricardo Feijoo Costa



This work is licensed under a [Creative Commons Attribution-ShareAlike 4.0 International License](https://creativecommons.org/licenses/by-sa/4.0/)

Proceso Arranque GNU/Linux	
BIOS or UEFI	MBR or GPT
Xestor Arranque	grub2: <e>,<Ctrl>+<x>
kernel	vmlinux + initrd ([gzip](cpio)),(squashfs)
init	sysvinit (/etc/inittab) or upstart (/etc/init/rc-sysinit.conf) or systemd (/etc/systemd/, /lib/systemd/)
runlevels	/etc/init.d, /etc/rcX.d, insserv or update-rc.d or chkconfig or systemctl init, telinit
/etc/rc.local	executed at the end of each multiuser runlevel
dmesg pstree -paA, ps -efjH	print or control the kernel ring buffer, display a tree of processes

## man 7 boot: Descrición xeral da secuencia de arranque.

- A secuencia de arranque varía de un sistema a outro pero pódese dividir basicamente nos seguintes pasos:
  1. Arranque do hardware. (BIOS)
  2. Cargador do sistema operativo(SO) (**MBR** ou **GPT**)
  3. Posta en marcha do núcleo (kernel) (**GRUBv2** -> /boot/grub -> /boot: **vmlinux-version, config-version, System.map-version, initrd.img-version**)
  4. init (tendencia actual systemd)
  5. Scripts de arranque (runlevels -> /etc/init.d -> /etc/rcX.d -> /etc/rc.local)

### 1. Arranque do hardware:

Botón encendido ou reset --> BIOS(MBR) ou UEFI(GPT) --> Recoñecemento e comprobación básica do hardware (POST) --> Elexir dispositivo de arranque (boot) --> Disco duro (CD, DVD, USB, Rede...) --> Cargador do SO en memoria --> Control ao dispositivo de arranque

### 2. Cargador do SO --> MBR (Sector 0 = 512B) --> Táboa de particións --> Partición activa --> GRUBv2 --> **chamada ao núcleo (kernel)**

### 3. Posta en marcha do núcleo (kernel) --> parámetros de arranque (man 7 bootparam) --> inicializa dispositivos mediante os drivers --> arranca swapper --> monta sistema de ficheiros raíz (/) --> arranca proceso 1 en espazo de usuario --> /sbin/init con parámetros que non foron manexados polo núcleo

### 4. init:

init --> sistemas sysvinit --> /etc/inittab

init --> sistemas upstart --> /etc/init/rc-sysinit.conf

**init (man 1 init) --> sistemas systemd (tendencia actual)--> /etc/systemd, /lib/systemd**

--> runlevels (niveis de execución) (man 8 runlevel) --> systemctl (man 1 systemctl) --> cada nivel asóciase cun conxunto de servizos (man 8 init || man 8 telinit)

### 5. Scripts de arranque:

- Para cada servizo --> 1 script de inicialización en /etc/init.d (start, stop, restart, status)
- Para runlevel X --> /etc/rcX.d --> servizos en orde numérica que poden comezar por S (start) ou K (stop), os cales son accesos directos (ligazóns simbólicas) a servizos en /etc/init.d
- Para modificar esas ligazóns simbólicas podemos empregar comandos específicos para tal fin: insserv, chkconfig, **update-rc.d** (en debian), **systemctl** (en sistemas systemd)
- Logo dos scripts de arranque execútase o script /etc/rc.local no caso que exista.

Ricardo Feijoo Costa



This work is licensed under a **Creative Commons Attribution-ShareAlike 4.0 International License**

# Comandos iniciais

**man, help, echo, cd, ls, mkdir, rmdir, rm, cp, ln, mv, pwd**

- **man:** Ver páxinas do manual dos comandos.

\$ **man man** #Ver o manual do comando man. Para saír das páxinas do manual premer a tecla **q**. Para buscar patróns dentro das páxinas premer a tecla **/** e a continuación escribir o patrón de búsqueda, é dicir, se queremos buscar o patrón man escribir **/man**. Para seguir buscando referencias do patrón buscado hacia adiante premer **n**, e **N** se se quere ir hacia atrás nas referencias atopadas.

A distribución inferior amosa os números de sección do manual seguidos polo tipo de páxina que contén:

- 1 Programa executable ou comandos da shell
- 2 Chamadas ao sistema (funcións proporcionados polo núcleo)
- 3 Chamadas a biblioteca (funcións dentro de bibliotecas de programa)
- 4 Ficheiros especiais (usualmente atopado en /dev)
- 5 Formatos de ficheiro e convencións p. e. /etc/passwd
- 6 Xogos
- 7 Diversas (inclúense paquetes macro e convencións), p. e. man(7), groff(7)
- 8 Comandos administrativos do sistema (normalmente so para administrador)
- 9 Rutinas do núcleo [Non comúns]

\$ **man ls** #Ver o manual do comando ls. Para buscar a opción **-l** premer **/-l**

\$ **man cp** #Ver o manual do comando cp. Para buscar a opción **-p** premer **/-p**

\$ **man 1 cp** #Ver ás páxinas do manual referidas á sección 1 do comando cp. Para buscar a opción **-p** premer **/-p**

\$ **man 8 sudo** #Ver ás páxinas do manual referidas á sección 8 do comando sudo.

\$ **man ln** #Ver o manual do comando ln. Para buscar a opción **-s** premer **/-s**

- **help:** Amosar información de uso (axuda) sobre os comandos internos (builtin). Estes comandos son internos á Shell Bash e están sempre cargados na súa memoria.

\$ **help help** #Amosar información de uso do comando interno help

\$ **help** #Amosar todos os comandos internos

\$ **help type** #Amosar información sobre o comando interno type

\$ **help cd** #Amosar información sobre o comando interno cd

\$ **help echo** #Amosar información sobre o comando interno echo

- **echo:** Escribe os argumentos pasados á saída estándar (pantalla).

\$ **echo Ola mundo** #Volca na pantalla Ola mundo

\$ **echo -n Ola mundo** #Volca na pantalla Ola mundo e non engade unha nova liña

\$ **echo -e 'Ola mundo\nMundo ola'** #Coa opción -e interpreta os caracteres escapados coa barra invertida '\'. Así, neste caso volca na pantalla Ola mundo, interpreta \n que significa nova liña, e volca a continuación Mundo ola

\$ **echo -e 'Ola mundo\nMundo ola\tOla mundo'** #Coa opción -e interpreta os caracteres escapados coa barra invertida '\'. Así, neste caso volca na pantalla Ola mundo, interpreta \n que significa nova liña, volca a continuación Mundo ola, interpreta \t que significa tabulado horizontal e volca de novo Ola mundo

- **cd:** Cambiar de directorio de traballo da SHELL.

\$ **cd /tmp** #Cambiar o directorio de traballo a /tmp, é dicir, acceder a /tmp

\$ **cd .** #Cambiar o directorio de traballo actual, é dicir, permanecer no directorio de traballo actual: o directorio punto '.'

\$ **cd ..** #Cambiar o directorio de traballo pai (o directorio punto punto '..'), é dicir, o inmediatamente anterior, neste caso o directorio raíz '/'

\$ **cd** #Cambiar o directorio \$HOME do usuario que executa o comando, é dicir, se

\$HOME=/home/usuario, cambiarase ao directorio /home/usuario

\$ cd - #Voltar ao directorio no que estabamos situados previo á ruta actual, neste caso voltará ao directorio raíz '/'

\$ cd ~ #Equivale a cd \$HOME ou ao comando cd, é dicir, voltar ao directorio de traballo do usuario.

\$ cd /tmp/.. #Acceder a /tmp e logo voltar ao directorio pai, é dicir, acceder ao directorio raíz '/'.

\$ cd /home/usuario/../../.. #Acceder a /home/usuario, e logo permanecer nese directorio, e logo voltar ao directorio pai, e logo permanecer nese directorio, e logo voltar ao directorio pai, e logo voltar ao directorio pai do anterior, é dicir, acceder ao directorio raíz '/'.

- ls: Listar o contido do directorio actual (ficheiros e directorios). A non ser que se empregue a opción -a non se listarán os ficheiros que comezan polo carácter punto (.), que son considerados como ficheiros ocultos

\$ ls -l #Listar o contido do directorio actual (ficheiros e directorios) de forma extendida, amosando os permisos do ficheiro, o número de ligazóns que ten, o nome do usuario propietario, o do grupo ao que pertence, o tamaño (en bytes), unha marca de tempo, e o nome do ficheiro. De forma predeterminada, a marca de tempo que se amosa é a da última modificación (mtime). Para ficheiros especiais de dispositivo o campo de tamaño sustitúese comúnmente polos números de dispositivo maior e menor. Non se listarán os ficheiros que comezan polo carácter punto (.), que son considerados como ficheiros ocultos

\$ ls -i #Listar o contido do directorio actual (ficheiros e directorios) precedendo a saída para o ficheiro polo número de serie do ficheiro (inodo). Non se listarán os ficheiros que comezan polo carácter punto (.), que son considerados como ficheiros ocultos

\$ ls -a #Listar o contido do directorio actual (ficheiros e directorios) amosando os ficheiros que comezan polo carácter punto (.), os cales son considerados como ficheiros ocultos

\$ ls -l -i -a #Executar o comando ls coas opcións -l, -i e -a

\$ ls -lia #Comando equivalente ao anterior. As opcións dos comandos pódense concatenar sen espazos a continuación dun carácter -

\$ ls -liahr #Executar o comando ls coas opcións -l, -i, -a, -h, -t e -r. A opción -h engade unha letra indicativa de tamaño, tal como M para megabytes binarios ('mebibytes'), a cada tamaño. A opción -t clasifica polo tempo de modificación (o 'mtime' no inodo) en vez de alfabeticamente, cos ficheiros máis recentes en primeiro lugar. A opción -r clasifica en orde inversa. Polo tanto, o comando lista ficheiros e directorios do directorio actual amosando de abaixo hacia arriba os máis recentes e en formato de lectura de tamaño máis amigable para as persoas (K, M, G...)

\$ ls -R #Listar o contido (ficheiros e directorios) do directorio actual e de todos os seus subdirectorios, é dicir, listar de forma recursiva dende o directorio actual. Non se listarán os ficheiros que comezan polo carácter punto (.), que son considerados como ficheiros ocultos

\$ ls /tmp #Listar o contido (ficheiros e directorios) do directorio /tmp. Non se listarán os ficheiros que comezan polo carácter punto (.), que son considerados como ficheiros ocultos

\$ ls -R /tmp #Listar o contido (ficheiros e directorios) do directorio /tmp e de todos os seus subdirectorios, é dicir, listar de forma recursiva dende o directorio /tmp. Non se listarán os ficheiros que comezan polo carácter punto (.), que son considerados como ficheiros ocultos

\$ ls -R / #Listar o contido (ficheiros e directorios) de todo o sistema operativo (/). Non se listarán os ficheiros que comezan polo carácter punto (.), que son considerados como ficheiros ocultos

- mkdir: Crear directorios.

\$ mkdir /tmp/dir1 #Crear o directorio dir1 dentro de /tmp

\$ mkdir /tmp/dir2 /tmp/dir3 /tmp/dir4 #Crear os directorios dir2, dir3, dir4 dentro de /tmp

\$ mkdir /tmp/dir1/dirA/dirB #Comando erróneo porque para crear dirB, debe existir dirA dentro de /tmp/dir1

\$ mkdir -p /tmp/dir1/dirA/dirB #Crear todos os directorios e subdirectorios ata chegar ao directorio final dirB, é dicir, crease dirB e todos os directorios que non existan para chegar a crear dirB

- **rmdir:** Eliminar directorios baleiros.

```
$ rmdir /tmp/dir4 #Eliminar o directorio /tmp/dir4
```

```
$ rmdir /tmp/dir1/dirA #Comando erróneo. Non se pode eliminar o directorio /tmp/dir1/dirA, porque non está baleiro, xa que contén o directorio dirB
```

```
$ cd /tmp #Cambiar o directorio de traballo a /tmp, é dicir, acceder a /tmp
```

```
$ rmdir -p ./dir1/dirA/dirB #Eliminar todos os directorios e subdirectorios da ruta indicada dentro de /tmp. Así, sería equivalente á facer: rmdir /tmp/dir1/dirA/dirB; rmdir /tmp/dir1/dirA; rmdir /tmp/dir1
```

```
$ mkdir -p /tmp/dir1/dirA/dirB #Crear todos os directorios e subdirectorios ata chegar ao directorio final dirB, é dicir, crease dirB e todos os directorios que non existan para chegar a crear dirB
```

```
$ rmdir -p dir1/dirA/dirB #Eliminar todos os directorios e subdirectorios da ruta indicada dentro de /tmp. Así, sería equivalente á facer: rmdir /tmp/dir1/dirA/dirB; rmdir /tmp/dir1/dirA; rmdir /tmp/dir1
```

```
$ mkdir -p /tmp/dir1/dirA/dirB #Crear todos os directorios e subdirectorios ata chegar ao directorio final dirB, é dicir, crease dirB e todos os directorios que non existan para chegar a crear dirB
```

```
$ rmdir -p /tmp/dir1/dirA #Comando erróneo. Non se pode eliminar o directorio /tmp/dir1/dirA, porque non está baleiro, xa que contén o directorio dirB
```

- **rm:** Eliminar ficheiros e directorios, incluso os directorios baleiros.

```
$ touch /tmp/1.tmp /tmp/2.tmp /tmp/3.tmp /tmp/4.tmp #Crear os arquivos /tmp/1.tmp, /tmp/2.tmp, /tmp/3.tmp e /tmp/4.tmp se non existen. Se existen modifica as datas de acceso, modificación e cambio.
```

```
$ rm /tmp/4.tmp #Eliminar o ficheiro /tmp/4.tmp
```

```
$ rm /tmp/2.tmp /tmp/3.tmp #Eliminar os ficheiros /tmp/2.tmp e /tmp/3.tmp
```

```
$ mkdir -p /tmp/dir1/dirA/dirB #Crear todos os directorios e subdirectorios ata chegar ao directorio final dirB, é dicir, crease dirB e todos os directorios que non existan para chegar a crear dirB
```

```
$ rm /tmp/dir1/dirA/dirB #Comando erróneo. Como dirB é un directorio, baleiro ou non, para poder borrarlo debemos empregar a opción -r, senón rm soamente eliminará ficheiros e non directorios
```

```
$ rm -r /tmp/dir1/dirA/dirB #Eliminar o directorio, baleiro ou non, dirB situado en /tmp/dir1/dirA
```

```
$ rm -r /tmp/dir1/ #Eliminar á árbore de directorios, baleiros ou non, a partir do directorio especificado, neste caso dir1, é dicir, borra todo (ficheiros e directorios) de forma recursiva a partir de /tmp/dir1, incluído dir1
```

```
$ mkdir -p /tmp/dir1/dirA/dirB #Crear todos os directorios e subdirectorios ata chegar ao directorio final dirB, é dicir, crease dirB e todos os directorios que non existan para chegar a crear dirB
```

```
$ rm -rf /tmp/dir1/ #Eliminar á árbore de directorios sen pedir confirmación (opción -f), baleiros ou non, a partir do directorio especificado, neste caso dir1, é dicir, borra todo (ficheiros e directorios) de forma recursiva a partir de /tmp/dir1, incluído dir1
```

- **cp:** Copiar ficheiros e directorios

```
$ cp /etc/passwd /tmp #Copiar o ficheiro orixe /etc/passwd no ficheiro destino /tmp/passwd
```

```
$ ls -l /etc/passwd /tmp/passwd #Listar en formato extendido os ficheiros /etc/passwd e /tmp/passwd
```

```
$ cp -pv /etc/passwd /tmp #Copiar o ficheiro orixe /etc/passwd no ficheiro destino /tmp/passwd, e coa opción -p preservando os permisos(ugo) e datas no ficheiro destino /tmp/passwd. E coa opción -v en modo detallado (verbose)
```

```
$ ls -l /etc/passwd /tmp/passwd #Listar en formato extendido os ficheiros /etc/passwd e /tmp/passwd
```

```
$ cp /etc/passwd /etc/group /tmp #Copiar os ficheiros orixe /etc/passwd e /etc/group no
```



directorio destino /tmp

\$ cp -Rv \$HOME /tmp #Copiar de forma recursiva o directorio \$HOME en /tmp, é dicir, se \$HOME=/home/usuario copiar todo /home/usuario en /tmp

- ln: Crear enlaces entre ficheiros

\$ cp /etc/passwd /tmp #Copiar o ficheiro orixe /etc/passwd no ficheiro destino /tmp/passwd

\$ cd /tmp #Cambiar o directorio de traballo a /tmp, é dicir, acceder a /tmp

\$ ln -s passwd PASSWD-SYMBOLIC #Crear na ruta actual a ligazón simbólica PASSWD-SYMBOLIC que apunta a /tmp/passwd

\$ ls -li passwd PASSWD-SYMBOLIC #Listar os ficheiros /tmp/passwd, PASSWD-SYMBOLIC de forma extendida precedendo as súas saídas polos números de serie dos ficheiros (inodos).

\$ ln passwd PASSWD-HARD #Crear na ruta actual a ligazón dura PASSWD-HARD que apunta a /tmp/passwd. Nesta caso non se crea un novo ficheiro, senón un novo nome de ficheiro que apunta ao mesmo inodo.

\$ ls -li passwd PASSWD-SYMBOLIC PASSWD-HARD #Listar os ficheiros /tmp/passwd, PASSWD-SYMBOLIC, PASSWD-HARD de forma extendida precedendo as súas saídas polos números de serie dos ficheiros (inodos).

- mv: Mover (renomear) ficheiros e directorios

\$ cp /etc/passwd /tmp #Copiar o ficheiro orixe /etc/passwd no ficheiro destino /tmp/passwd

\$ mv /tmp/passwd /tmp/passwd1 #Mover o ficheiro /tmp/passwd a /tmp/passwd1. Como movemos á mesma ruta en realidade estamos a renomear o ficheiro.

\$ mkdir -p /tmp/directorio1/directorio2/directorio3 #Crear todos os directorios e subdirectorios ata chegar ao directorio final directorio3, é dicir, crease directorio3 e todos os directorios que non existan para chegar a crear directorio3

\$ mv /tmp/passwd1 /tmp/directorio1/directorio2/ #Mover o ficheiro /tmp/passwd1 dentro do directorio /tmp/directorio1/directorio2

\$ mv /tmp/directorio1/directorio2 /tmp #Mover o directorio /tmp/directorio1/directorio2 e todo o seu contido a /tmp

\$ touch /tmp/directorio1/ff1.tmp /tmp/directorio1/ff2.tmp /tmp/directorio1/ff3.tmp /tmp/directorio1/ff4.tmp #Crear os arquivos /tmp/directorio1/ff1.tmp, /tmp/directorio1/ff2.tmp, /tmp/directorio1/ff3.tmp e /tmp/directorio1/ff4.tmp se non existen. Se existen modifica as datas de acceso, modificación e cambio.

\$ mv /tmp/directorio1/ff?.tmp /tmp/directorio2/directorio3 #Mover a /tmp/directorio2/directorio3 os ficheiros existentes na ruta /tmp/directorio1 onde o nome comeza por 3 caracteres, os 2 primeiros son ff e o terceiro, que debe existir, pode ser calquera caracter, e que acabe cos caracteres .tmp

\$ mv /tmp/directorio2/directorio3/\*.tmp /tmp/directorio1 #Mover a /tmp/directorio1 os ficheiros existentes na ruta /tmp/directorio2/directorio3 onde o nome comeza por ningún caracter ou calquera número de caracteres sempre e cando remate cos caracteres .tmp

- pwd: Amosar o nome do directorio de traballo actual

\$ cd ~ #Equivale a cd \$HOME ou ao comando cd, é dicir, voltar ao directorio de traballo do usuario.

\$ pwd #Devolver o nome do directorio actual. Neste caso o directorio de traballo do usuario

\$ cd /tmp #Acceder ao directorio /tmp

\$ pwd #Devolver o nome do directorio actual. Neste caso o directorio /tmp

Ricardo Feijoo Costa



This work is licensed under a **Creative Commons Attribution-ShareAlike 4.0 International License**

# Comandos básicos

**cat, head, tail, tac, nl, wc, less, more, type, diff, touch, md5sum, sha512sum**

- cat: Ver contido de ficheiros.

```
$ cat /etc/passwd #Ver o contido do ficheiro /etc/passwd
```

```
$ cat /etc/group #Ver o contido do ficheiro /etc/group
```

```
$ cat -n /etc/group #Ver o contido do ficheiro /etc/group numerando as liñas
```

- head: Por defecto ver o contido das 10 primeiras liñas dun ficheiro.

```
$ head /etc/passwd #Ver o contido das 10 primeiras liñas do ficheiro /etc/passwd
```

```
$ head -n 8 /etc/passwd #Ver o contido das 8 primeiras liñas do ficheiro /etc/passwd
```

```
$ head -8 /etc/passwd #Igual que o comando anterior
```

- tail: Por defecto ver o contido das 10 últimas liñas dun ficheiro.

```
$ tail /etc/passwd #Ver o contido das 10 últimas liñas do ficheiro /etc/passwd
```

```
$ tail -n 8 /etc/passwd #Ver o contido das 8 últimas liñas do ficheiro /etc/passwd
```

```
$ tail -8 /etc/passwd #Igual que o comando anterior
```

```
$ tail -f /etc/passwd #Deixar aberto o ficheiro /etc/passwd para lectura, comenzando a ver polas 10 últimas liñas.
```

```
$ tail -n 2 -f /etc/passwd #Deixar aberto o ficheiro /etc/passwd para lectura, comenzando a ver polas 2 últimas liñas.
```

```
$ tail -f /var/log/syslog #Deixar aberto o ficheiro /var/log/messages para lectura, comenzando a ver polas 10 últimas liñas. Moi empregado na revisión de logs. Neste caso como o usuario non ten permisos de lectura sobre o ficheiro a execución do comando amosará un erro.
```

```
$ tail -n 2 -f /var/log/syslog #Deixar aberto o ficheiro /var/log/messages para lectura, comenzando a ver polas 2 últimas liñas. Moi empregado na revisión de logs. Neste caso como o usuario non ten permisos de lectura sobre o ficheiro a execución do comando amosará un erro.
```

```
$ sudo su - #Acceder como root como se fixera login sen sudo. Comúmente empregado nas ISO LIVE de GNU/Linux para convertirse en root.
```

```
# tail -f /var/log/syslog #Deixar aberto o ficheiro /var/log/messages para lectura, comenzando a ver polas 10 últimas liñas. Moi empregado na revisión de logs
```

```
# tail -n 2 -f /var/log/syslog #Deixar aberto o ficheiro /var/log/messages para lectura, comenzando a ver polas 2 últimas liñas. Moi empregado na revisión de logs
```

```
# tail -2 -f /var/log/syslog #Comando erróneo xa que a opción -2 esta a ser empregado nun contexto inválido, ao empregar tamén a opción -f
```

```
# exit #Pechar a consola de comandos do usuario actual, neste caso do usuario root
```

- tac: O inverso de cat, é dicir, ver o contido dun ficheiro empezando dende a última liña ata a primeira.

```
$ tac /etc/passwd #Ver o contido do ficheiro /etc/passwd empezando dende a última liña ata a primeira
```

```
$ tac -n /etc/passwd #Erro. Non existe a opción -n
```

- nl: Ver o contido dos ficheiros numerando as liñas.

```
$ nl /etc/passwd #Ver o contido do ficheiro /etc/passwd numerando as liñas
```

- wc: Imprimir o número de liñas, palabras, caracteres e bytes dos ficheiros.

```
$ wc -l /etc/passwd #Imprimir o número de liñas do ficheiro /etc/passwd
```

```
$ wc -m /etc/passwd #Imprimir o número de caracteres do ficheiro /etc/passwd
```

\$ wc -c /etc/passwd #Imprimir o reconto de bytes do ficheiro /etc/passwd

\$ wc -w /etc/passwd #Imprimir o número de palabras do ficheiro /etc/passwd

- less: Ver contido dun ficheiro podendo moverse por el a través das teclas frechas (←, ↑, →, ↓), ou <AvPág> e <RePág>

\$ less /etc/passwd

- more: Ver contido dun ficheiro paxinado facendo pausa según as dimensións da pantalla. Non permite retroceder para voltar a visionar parte do arquivo mostrado.

\$ more /etc/passwd

- diff: Compara 2 arquivos amosando as súas diferencias.

- cp -pv /etc/passwd /tmp/passwd

- diff /etc/passwd /tmp/passwd

- diff /etc/passwd /etc/group

- touch: Permite modificar datas(acceso, modificación, cambio) de arquivos . Ademais permite xerar ficheiros baleiros se non existen.

\$ touch /tmp/a.tmp #Crea o arquivo /tmp/a.tmp se non existe. Se existe modifica as datas de acceso, modificación e cambio.

- md5sum: Crea hash MD5.

\$ md5sum /etc/passwd #Crea hash MD5 do ficheiro /etc/passwd

- sha512sum: Crea hash SHA512.

\$ sha512sum /etc/passwd #Crea hash SHA512 do ficheiro /etc/passwd

Ricardo Feijoo Costa



This work is licensed under a **Creative Commons Attribution-ShareAlike 4.0 International License**



# Metacaracteres: ?, \*, [...]

**? (1 caracter)**

**\* (zero ou varios caracteres)**

**[...] (algún dos caracteres indicados)**

## Escenario

A maiores dos que poidan existen no cartafol /tmp vanse crear os seguintes ficheiros:

```
$ touch /tmp/proba1 /tmp/proba2 /tmp/proba12 /tmp/proba23 /tmp/probA1 /tmp/probA2 /tmp/probA12 /tmp/probA23 #Creación de ficheiros baleiros en /tmp
```

## Práctica

- ?: 1 caracter.

```
$ ls /tmp/proba? #Lista /tmp/proba1, /tmp/proba2
```

```
$ ls /tmp/proba?? #Lista /tmp/proba12, /tmp/proba23
```

```
$ ls /tmp/proba??? #Non lista ningún dos ficheiros creados anteriormente
```

```
$ ls /tmp/probA? #Lista /tmp/probA1, /tmp/probA2
```

```
$ ls /tmp/probA?? #Lista /tmp/probA12, /tmp/probA23
```

```
$ ls /tmp/probA??? #Non lista ningún dos ficheiros creados anteriormente
```

```
$ ls /tmp/prob?? #Lista /tmp/proba1, /tmp/probA1, /tmp/proba2, /tmp/probA2
```

```
$ ls /tmp/prob??? #Lista /tmp/proba12, /tmp/probA12, /tmp/proba23, /tmp/probA23
```

```
$ ls /tmp/prob???? #Non lista ningún dos ficheiros creados anteriormente
```

```
$ ls /tmp/?proba1 #Non lista ningún dos ficheiros creados anteriormente
```

- \*: 0 ou varios caracteres.

```
$ ls /tmp/prob* #Lista todos os ficheiros creados anteriormente
```

```
$ ls /tmp/*ro* #Lista todos os ficheiros creados anteriormente
```

```
$ ls /tmp/p* #Lista todos os ficheiros creados anteriormente
```

\$ ls /tmp/\*b\* #Lista todos os ficheiros criados anteriormente

\$ ls /tmp/proba\* #Lista /tmp/proba1, /tmp/proba12, /tmp/proba2, /tmp/proba23

\$ ls /tmp/probA\* #Lista /tmp/probA1, /tmp/probA12, /tmp/probA2, /tmp/probA23

\$ ls /tmp/\*A\* #Lista /tmp/probA1, /tmp/probA12, /tmp/probA2, /tmp/probA23

\$ ls /tmp/\*p\* #Lista todos os ficheiros criados anteriormente

\$ ls /tmp/\*probA23\* #Lista /tmp/probA23

- [...]: algún dos caracteres indicados.

\$ ls /tmp/[prob] #Non lista ningún dos ficheiros criados anteriormente

\$ ls /tmp/[prob\*] #Non lista ningún dos ficheiros criados anteriormente

\$ ls /tmp/prob[aA]1 #Lista /tmp/proba1, /tmp/probA1

\$ ls /tmp/prob[aA][12] #Lista /tmp/proba1, /tmp/probA1, /tmp/proba2, /tmp/probA2

\$ ls /tmp/prob[aA][123] #Lista os mesmos que o comando anterior: /tmp/proba1, /tmp/probA1, /tmp/proba2, /tmp/probA2

\$ ls /tmp/prob[aA][12][3] #Lista /tmp/proba23, /tmp/probA23

\$ ls /tmp/[pr]ob[aA][12][3] #Non lista ningún dos ficheiros criados anteriormente

- Combinados:

\$ ls /tmp/??[rR]o\* #Non lista ningún dos ficheiros criados anteriormente

\$ ls /tmp/?[rR]o\* #Lista todos os ficheiros criados anteriormente

\$ ls /tmp/prob[aA]\* #Lista todos os ficheiros criados anteriormente

\$ ls /tmp/prob?\* #Lista todos os ficheiros criados anteriormente

\$ ls /tmp/prob?[aA]\* #Non lista ningún dos ficheiros criados anteriormente

Ricardo Feijoo Costa



This work is licensed under a **Creative Commons Attribution-ShareAlike 4.0 International License**

# PROMPTS

## echo \$PS<Tab><Tab>

- Son variables de entorno
  - Tipos:
    - PS0--> Prompt para shells interactivos. Amosa unha mensaxe logo de ler un comando e antes da súa execución.
    - PS1--> Prompt primario --> Normalmente indica:  
`usuario@maquina:cartafol_actual$`
- Modifícase permanentemente en:
- `~/.bashrc` (`~`  $\Rightarrow$  `$HOME`) para un usuario particular
  - `/etc/bash.bashrc` para tódolos usuarios.
- PS2--> Prompt secundario --> Por defecto vale `>`. Emprégase para o modo entrada multiliña.
  - PS3--> Prompt de programación --> Valor por defecto `#?`. Emprégase en select para os elementos de control de selección.
  - PS4--> Prompt de depuración --> Por defecto `+`. Aparece ao comezo de cada liña namentras estase depurando o script (`bash -x` ou `set -x`)
- Opcións de prompt--> Caracteres de escape: `LANG=en_US man bash` (buscar PROMPTING).

## Práctica

1. Abrir terminal e executar os seguintes comandos:

```
alumno@debian:~$ echo $PS<Tab><Tab>
$PS1 $PS2 $PS4
alumno@debian:~$ echo $PS0

alumno@debian:~$ echo $PS1
\[e]0;\u@\h: \w\a\>${debian_chroot:+($debian_chroot)}\u@\h:\w\$
alumno@debian:~$ echo $PS2
>
alumno@debian:~$ echo $PS3

alumno@debian:~$ echo $PS4
+
```

2. Modificar prompt PS0:

```
alumno@debian:~$ PS0='Executando... \n'
alumno@debian:~$ echo Ola mundo
```

3. Modificar prompt PS1:

```
alumno@debian:~$ PS1='Ola mundo! -Por ser orixinais- >> '
```

4. Modificar prompt PS1 coas opcións: `\d:\T:\l:\!`

```
alumno@debian:~$ PS1='${debian_chroot:+($debian_chroot)}\u@\h:\w\d:\T:\l:\! $ '
```

5. Modificar prompt PS1 coa opción `\D` (ver *man strftime* para os seus argumentos):  
`\D{%A %e %B %G %T}:\!`

```
alumno@debian:~$ PS1='${debian_chroot:+($debian_chroot)}\u@\h:\w \D{%A %e %B %G %T}:\! $ '
```

Ricardo Feijoo Costa



This work is licensed under a [Creative Commons Attribution-ShareAlike 4.0 International License](https://creativecommons.org/licenses/by-sa/4.0/)

# Variables e comiñas, comiñas e variables

## Crear variables

Para crear unha variable soamente temos que identificala cun nome e igualala ao seu valor, por exemplo:

# Caracteres backticks ou comiñas invertidas

```
IP=`/sbin/ifconfig eth0`
```

##### Alternativa a estas comiñas: \$()

```
IP=$(/sbin/ifconfig eth0)
```

# Comiñas simples

```
RUTA_PRINCIPAL='/home/alumno'
```

# Comiñas dobres

```
NOME="usuario"
```

```
RUTA_SECUNDARIA="$RUTA_PRINCIPAL/tmp"
```

- **Definición simplificada:** As variables en Bash se definen como NOME=valor (sen espazos antes ou despois do símbolo =) e o seu valor emprégase, poñendo o símbolo \$ diante do nome da variable, \$NOME.
- **Definición xenérica:** NOME=\${valor}.
- A forma xenérica debe empregarse sempre que a variable vaia seguida de letra, dígito ou guión baixo; noutro caso pode empregarse a forma simplificada.

## Práctica

### EXEMPLO A:

```
$ NOME='Ricardo'
$ APELIDO1='Feijoo'
$ APELIDO2='Costa'
$ echo $NOME_$(APELIDO1)_$(APELIDO2)
```

A execución deste exemplo devolve a seguinte saída na consola:

```
Costa
```

### EXEMPLO B:

```
$ NOME='Ricardo'
$ APELIDO1='Feijoo'
$ APELIDO2='Costa'
$ echo ${NOME}_${APELIDO1}_${APELIDO2}
```

A execución deste exemplo devolve a seguinte saída na consola:

```
Ricardo_Feijoo_Costa
```

Ricardo Feijoo Costa



This work is licensed under a [Creative Commons Attribution-ShareAlike 4.0 International License](https://creativecommons.org/licenses/by-sa/4.0/)

# SHELL BASH: /bin/bash

## echo \$SHELL

- Orde de preferencia: 5 tipos

alias --> keywords --> functions --> **builtin** --> file

onde:

*alias* --> \$HOME/.bashrc, /etc/bashrc

*keywords* --> palabras clave: function, if, for...

*functions* --> funcións: nome\_funcion() {...}

*builtin* --> comandos internos: cd, type... Son internos a Bash e están sempre cargados na súa memoria.

*file* --> scripts e programas executables (según PATH)

## Práctica

- `type <nome_comando>`: amosa información sobre o tipo de comando
- `type type`: amosa información sobre o comando type
- `help`: Amosa a axuda dos comandos internos, xa que o man non a posúe.
  - \$ `help help` #amosa información sobre o comando interno help
  - \$ `help cd`
  - \$ `help #` amosa todos os comandos internos
  - \$ `help type` #amosa información sobre o comando interno type
- `type -t`: amosa o tipo do/s comando/s pasado/s como argumento/s
  - \$ `type -t ls`
- `type -a`: amosa tódolos valores nos tipos do/s comando/s pasado/s como argumento/s, é dicir, tódalas posibilidades de execución do/s comando/s.
  - \$ `type -a ls`
- **command**: Na súa execución soamente busca comandos builtin e file
  - \$ `type -t ls`
  - \$ `type -a ls`
  - \$ `alias ls='echo'`
  - \$ `type -t ls`
  - \$ `type -a ls`
  - \$ `ls`
  - \$ `command ls`
  - \$ `unalias ls`
  - \$ `ls`
- **builtin**: Na súa execución soamente busca comandos internos (builtin)
  - \$ `type -t cd`
  - \$ `type -a cd`
  - \$ `cd /tmp`
  - \$ `builtin cd`
  - \$ `alias cd='echo'`

```
$ type -t cd
$ type -a cd
$ cd
$ cd /tmp
$ builtin cd /tmp
$ unalias cd
```

- enable: Permite activar/desactivar comandos internos.

```
$ type -t enable
$ type -a enable
$ help enable
$ enable -a
$ enable -n
$ enable -n cd
$ enable -n
$ enable -a
$ cd /tmp
$ builtin cd /tmp
$ type -t cd
$ type -a cd
$ enable cd
$ cd /tmp
```

---

[Ricardo Feijoo Costa](#)



This work is licensed under a [Creative Commons Attribution-ShareAlike 4.0 International License](#)



# Historial de comandos

- history: Ver historial de comandos
- ~/.bash\_history: Ficheiro onde se gardar os comandos executados feito logout.
- fc : Ver últimos comandos e editar un deles.
- !! : Executar último comando
- !nº : Executar comando número nº
- !comezo\_comando : Executar último comando que comeze pola cadea comezo\_comando
- <Ctrl>+<r> : Buscar no historial

## Práctica

Abrir terminal e executar os seguintes comandos:

```
$ type -t history
$ help history
$ history
$ history -c
$ history
$ echo $HIST<Tab>
$ man bash (Buscar HISTSIZE e HISTFILESIZE)
$ help fc
$ fc -l
$ fc -l -n
$ fc nº_comando
$ ls
$ !!
$ !his<Enter>
$ id
$ groups
$ !!
$ <Ctrl>+<r>his<Enter>
```

Ricardo Feijoo Costa



This work is licensed under a [Creative Commons Attribution-ShareAlike 4.0 International License](https://creativecommons.org/licenses/by-sa/4.0/)

# Entrada e Saída: Redireccións e tuberías

**STDIN (0), STDOUT(1), STDERR(2), >, >>, 2>, 2>>, <, cat > filename <<EOF...EOF, |, tee**

- STDIN (0): Entrada estándar teclado
- STDOUT (1): Saída estándar consola
- STDERR (2): Saída estándar consola
- > : redirixir STDOUT a un ficheiro
- >> : redirixir STDOUT ao final dun ficheiro engadindo se o ficheiro existe
- 2> : redirixir STDERR a un ficheiro. ¡OLLO! Non existe espazo entre o número 2 e o signo maior
- 2>> : redirixir STDERR ao final dun ficheiro engadindo se o ficheiro existe. ¡OLLO! Non existe espazo entre o número 2 e os signos maior
- < : redirixir a entrada dun comando, ou ben, redirixir o contido dun ficheiro a STDIN
- cat > filename <<EOF...EOF : redirixir a entrada do escrito entre os EOF ao ficheiro filename. Moi empregado en scripts
- Tuberías | : comandoN | comandoM --> Recoller a saída do comandoN (antes da tubería) como entrada para o comandoM (logo da tubería), é dicir, filtrar mediante o comandoM a saída do comandoN.
- tee : copia a entrada estándar á saída estándar e ao ficheiros especificados,

## Práctica

- STDIN (0): Entrada estándar teclado
- STDOUT (1): Saída estándar consola
- STDERR (2): Saída estándar consola
- > : redirixir STDOUT a un ficheiro

```
$ echo $HOME
$ su -
# echo $HOME
# exit
$ ls /tmp > $HOME/tmp.txt
$ cat $HOME/tmp.txt
$ /sbin/ifconfig > $HOME/ifconfig.txt
$ cat $HOME/ifconfig.txt
```
- >> : redirixir STDOUT ao final dun ficheiro engadindo se o ficheiro existe

```
$ ls /tmp > $HOME/tmp.txt
$ cat $HOME/tmp.txt
$ /sbin/ifconfig >> $HOME/tmp.txt
$ cat $HOME/tmp.txt
```
- 2> : redirixir STDERR a un ficheiro. ¡OLLO! Non existe espazo entre o número 2 e o signo maior

```
$ ls /tmpa
$ ls /tmpa 2> $HOME/tmp_errs.txt
```
- 2>> : redirixir STDERR ao final dun ficheiro engadindo se o ficheiro existe. ¡OLLO! Non existe espazo entre o número 2 e os signos maior

```
$ ls /tmpa 2> $HOME/tmp_errs.txt
$ /sbin/ifconfig 2>> $HOME/tmp_errs.txt
```
- < : redirixir a entrada dun comando, ou ben, redirixir o contido dun ficheiro a STDIN

```
$ cat < $HOME/tmp.txt
$ tr e E < $HOME/ifconfig.txt
```

- `cat > filename <<EOF...EOF` : redirixir a entrada do escrito entre os EOF ao ficheiro filename. Comunmente empregado en scripts.  
\$ `cat > /tmp/output.txt <<EOF`  
`cat /etc/passwd | grep root`  
`/sbin/ifconfig`  
`EOF`  
\$ `cat /tmp/output.txt`
- Tuberías | : `comandoN | comandoM -->` Recoller a saída do comandoN (antes da tubería) como entrada para o comandoM (logo da tubería), é dicir, filtrar mediante o comandoM a saída do comandoN.  
\$ `history | tail -20`  
\$ `cat $HOME/ifconfig.txt | tr eth ETH`
- `tee` : copia a entrada estándar á saída estándar e ao ficheiros especificados,  
\$ `/sbin/ifconfig | tee $HOME/ifconfig_tee.txt`  
\$ `ls /tmp | tee -a $HOME/ifconfig_tee.txt`

[Ricardo Feijoo Costa](#)



This work is licensed under a [Creative Commons Attribution-ShareAlike 4.0 International License](#)

# cut, awk, sed, tr, find, -exec, xargs

## Práctica

- cut: Corta seccións(bytes, caracteres ou campos) por cada liña na que actúa nun ficheiro ou sobre a entrada estándar. Por defecto o campo delimitador é o carácter de tabulación. Se se emprega un campo delimitador distinto, soamente pode ser un carácter, e será o empregado na saída para delimitar os campos.
- awk: Modifica patróns de texto orixe por patróns de texto destino e é tamén unha linguaxe de programación.
- sed: Modifica patróns de texto orixe por patróns de texto destino.
- tr: Modifica patróns de texto orixe por patróns de texto destino, carácter a carácter.
- find: Búsqueda de ficheiros e directorios.
- -exec: Opción de find que permite realizar comandos sobre o atopado na procura con find.
- xargs: Permite construír listas de parámetros e realizar comandos sobre as listas xeradas derivadas da execución doutros comandos.

## Práctica

```
$ rm -f /tmp/users.csv /tmp/Users.csv
```

**Eliminar de xeito forzoso(ignorando avisos e se existen ou non) os ficheiros /tmp/users.csv e /tmp/Users.csv**

```
$ cat > /tmp/users.csv <<EOF
"Usuario","Contraseña","Shell","Directorio de entrada ao sistema"
"user1","p1","/bin/bash","/tmp"
"user2","p2","/bin/false","/home/user2"
"user3","p3","/bin/bash","/home/user3"
"user4","p4","/bin/bash","/tmp"
"user5","p5","/bin/bash","/tmp"
"user6","p6","/bin/false","/home/user6"
"user7","p7","/bin/bash","/tmp"
"user8","p8","/bin/bash","/home/user8"
"user9","p9","/bin/false","/tmp"
"user10","p10","/bin/bash","/tmp"
EOF
```

**Crear ficheiro users.csv en /tmp**

```
$ cd /tmp
```

**Acceder ao cartafol /tmp**

```
$ cat users.csv
```

**Amosar contido do ficheiro users.csv**

```
$ cut -f1 users.csv
```

**Co -f1 querease amosar a primeira columna sendo o campo de separación o tabulado, xa que non existe no comando o parámetro -d. Entón, o que acontece é que como non existen caracteres de tabulación, amósase todo o contido do ficheiro.**

```
$ cat users.csv | cut -f1
```

**Comando equivalente ao anterior.**

```
$ cut -d, -f1 users.csv
```

**Existe o parámetro -d, entón o campo de separación é o carácter ','. Agora soamente se amosa a primeira columna sendo a separación entre columnas o carácter ','**

```
$ cat users.csv | cut -d, -f1
```

**Comando equivalente ao anterior.**

```
$ cut -d, -f1,2 users.csv
```

**Campo de separación: o carácter ','. Amósase a columna1 e a columna2 separadas polo carácter ','**

```
$ cut -d, -f2,1 users.csv
```

**Comando equivalente ao anterior, é dicir, non amosa primeiro a columna2 e logo a columna1, senón que amosa primeiro a columna1 e logo a columna2 separadas polo carácter ','**

```
$ cat users.csv | cut -d, -f1,2
```

**Comando equivalente ao anterior.**

```
$ cut -d, -f1,3 users.csv
```

**Campo de separación: o carácter ','. Amósase a columna1 e a columna3 separadas polo carácter ','**

```
$ cut -d, -f1-3 users.csv
```

**Campo de separación: o carácter ','. Amósase as columnas dende a columna1 ata a columna3 separadas polo carácter ','**

```
$ cut -d, -f3 users.csv
```

**Comando equivalente ao anterior.**

```
$ cut -d, -f3- users.csv
```

**Campo de separación: o carácter ','. Amósase as columnas dende a columna3 ata o final do ficheiro separadas polo carácter ','**

```
$ cut -c1-10 users.csv
```

**Amósase os primeiros 10 caracteres do ficheiro users.csv**

```
$ cut -d, -f3-1 users.csv
```

**Comando erróneo porque o rango solicitado é decrecente.**

```
$ cut -c -f10-1 users.csv
```

**Comando erróneo porque o rango solicitado é decrecente.**

```
$ awk '{print $1}' users.csv
```

**Co print \$1 quérese amosar a primeira columna sendo o campo de separación o espazo ou o tabulado, xa que non existe no comando o parámetro -F. Entón, o que acontece é que como non existen espazos, nin tabulados, amósase todo o contido do ficheiro.**

```
$ cat users.csv | awk '{print $1}'
```

**Comando equivalente ao anterior.**

```
$ awk -F, '{print $1}' users.csv
```

**Existe o parámetro -F, entón o campo de separación é o carácter ','. Agora soamente se amosa a primeira columna sendo a separación entre columnas o carácter ','.**

```
$ cat users.csv | awk -F, '{print $1}'
```

**Comando equivalente ao anterior.**

```
$ awk -F, '{print $1 " " $2}' users.csv
```

**Campo de separación: o carácter ','. Amósase a columna1 e a columna2 separadas polo carácter espazo.**

```
$ cat users.csv | awk -F, '{print $1 " " $2}'
```

**Comando equivalente ao anterior.**

```
$ sed 's//g' users.csv
```

**Sustituir o carácter comiñas dobres por ningún carácter, é dicir, eliminar o carácter comiñas dobres amosando a substitución por pantalla pero sen modificar o ficheiro users.csv.**

```
$ sed 's|'|g' users.csv
```

**Equivale ao comando anterior, soamente que a separación entre substitucións de patróns faise co carácter tubería(|) e non co carácter barra(/)**

```
$ sed 's/user/usuario/g' users.csv
```

**Sustituir user por usuario amosando a substitución por pantalla pero sen modificar o ficheiro users.csv.**

```
$ cat users.csv | sed 's/user/usuario/g'
```

**Comando equivalente ao anterior.**

```
$ sed -e 's/user/usuario/g' -e 's/\/home/\/tmp/g' users.csv
```

**Sustituir user por usuario e tamén /home por /tmp amosando a substitución por pantalla pero sen modificar o ficheiro users.csv.**

```
$ cat users.csv | sed -e 's/user/usuario/g' -e 's/\/home/\/tmp/g'
```

**Comando equivalente ao anterior.**

```
$ cat users.csv | sed -e 's/user/usuario/g' -e 's|/home|/tmp|g'
```

**Comando equivalente ao anterior.**

```
$ cp -pv users.csv Users.csv
```

**Copiar en modo verbose e preservando permisos(ugo), usuario, grupo e data o ficheiro users.csv a Users.csv.**

```
$ ls -l users.csv Users.csv
```

**Listar en modo extendido os ficheiros users.csv e Users.csv. Pódese comprobar como teñen os mesmos permisos(ugo), usuario, grupo e data.**

```
$ sed -i -e 's/user/usuario/g' -e 's/\/home/\/tmp/g' users.csv
```

**Sustituir user por usuario e tamén /home por /tmp sen amosar a substitución por pantalla e modificando o ficheiro users.csv.**

```
$ tr 'o' '0' < /tmp/users.csv
```

**Sustituir o carácter 'o' por '0' en /tmp/users.csv sen modificar o ficheiro e amosando a substitución pon pantalla.**

```
$ cat /tmp/users.csv | tr 'o' '0'
```

Igual que o comando anterior.

```
$ tr -s ' ' < /tmp/users.csv
```

Sustituir o carácter repetido ' '(espazo) por un só, ou sexa, sustitúe calquera número de espazos contiguos por un só espazo, sen modificar o ficheiro e amosando a substitución pon pantalla.

```
$ tr -d '\n' < /tmp/users.csv
```

Elimina o carácter retorno de carro (nova liña) sen modificar o ficheiro e amosando a substitución pon pantalla.

```
$ tr '\n' ' ' < /tmp/users.csv
```

Sustituir o carácter retorno de carro (nova liña) por un carácter ' '(espazo), sen modificar o ficheiro e amosando a substitución pon pantalla.

```
$ find . -name users.csv 2>/dev/null
```

Atopar na ruta actual calquera ficheiro/directorio de nome users.csv redireccionando calquera erro ao dispositivo /dev/null. Atópase user.csv e non Users.csv xa que o parámetro -name distingue entre caracteres maiúsculas e minúsculas.

```
$ find . -iname users.csv 2>/dev/null
```

Atopar na ruta actual calquera ficheiro/directorio de nome users.csv, non distinguindo entre maiúsculas e minúsculas redireccionando calquera erro ao dispositivo /dev/null. É dicir, atopa users.csv como Users.csv

```
$ find . -type f -exec grep -Hi "usuario" {} \; 2>/dev/null
```

Atopar na ruta actual calquera concurrencia do patrón a buscar 'usuario', en ficheiros, non distinguindo maiúsculas e minúsculas no patrón buscado e redireccionando calquera erro ao dispositivo /dev/null. É dicir, busca o patrón usuario nos ficheiros users.csv e Users.csv. A opción -H de grep permite no caso de atopar o patrón buscado imprimir o ficheiro no cal foi atopado.

```
$ find /tmp -type f -iname "u*.csv" -exec grep -Hi "usuario" {} \; 2>/dev/null
```

Atopar dentro da ruta /tmp, calquera ficheiro, que posúa como nome, independente de maiúsculas e minúsculas, empezando polo carácter 'u' e rematando cos caracteres '.csv', contendo entre os caracteres descritos anteriormente calquera número de caracteres (cero ou máis), para acabar buscando calquera concurrencia do patrón 'usuario', non distinguindo maiúsculas e minúsculas no patrón buscado. Redirecciona calquera erro ao dispositivo /dev/null. É dicir, busca o patrón usuario nos ficheiros users.csv e Users.csv. A opción -H de grep permite no caso de atopar o patrón buscado imprimir o ficheiro no cal foi atopado.

```
$ mkdir -p /tmp/a1/a2
```

Crear a estrutura de cartafoles /tmp/a1/a2 aínda que ningún dos cartafoles exista.

```
$ cp -pv /tmp/users.csv /tmp/a1/a2
```

Copiar en modo verbose e preservando permisos(ugo), usuario, grupo e data o ficheiro /tmp/users.csv á ruta /tmp/a1/a2

```
$ find / -maxdepth 1 -iname "u*.csv" -exec grep -Hi "usuario" {} \; 2>/dev/null
```

Atopar dentro da ruta /, pero soamente profundizando no nivel 1 de cartafoles, sendo o nivel 1, o contido do propio cartafol, nesta caso o cartafol raíz / (ou sexa, dentro de / os ficheiros e cartafoles que se visualizan executando o comando ls sen opcións).

Dentro do nivel 1 calquera arquivo (ficheiro, cartafol, link...) que posúa como nome, independente de maiúsculas e minúsculas, empezando polo carácter 'u' e rematando cos caracteres '.csv', contendo entre os caracteres descritos anteriormente calquera número de caracteres (cero ou máis), para acabar buscando calquera concurrencia do patrón 'usuario', non distinguindo maiúsculas e minúsculas no patrón buscado. Ademais redirecciona calquera erro ao dispositivo /dev/null. É dicir, non busca dentro de users.csv e Users.csv posto que co maxdepth 1 non os atopa. A opción -H de grep permite no caso de atopar o patrón buscado imprimir o ficheiro no cal foi atopado.

```
$ find / -maxdepth 1 -type f -iname "u*.csv" -exec grep -Hi "usuario" {} \; 2>/dev/null
```

Atopar dentro da ruta /, pero soamente profundizando no nivel 1 de cartafoles, sendo o nivel 1, o contido do propio cartafol, nesta caso o cartafol raíz / (ou sexa, dentro de / os ficheiros e cartafoles que se visualizan executando o comando ls sen opcións).

Dentro do nivel 1 non busca cartafoles, links, etc, soamente ficheiros, que posúan como nome, independente de maiúsculas e minúsculas, empezando polo carácter 'u' e rematando cos caracteres '.csv', contendo entre os caracteres descritos anteriormente calquera número de caracteres (cero ou máis), para acabar buscando calquera concurrencia do patrón 'usuario', non distinguindo maiúsculas e minúsculas no patrón buscado. Ademais redirecciona calquera erro ao dispositivo /dev/null. É dicir, non busca dentro de users.csv e Users.csv posto que co maxdepth 1 non os atopa. A opción -H de grep permite no caso de atopar o patrón buscado imprimir o ficheiro no cal foi atopado.

```
$ find / -maxdepth 2 -type f -iname "u*.csv" -exec grep -Hi "usuario" {} \; 2>/dev/null
```

Igual que o comando anterior pero profundizando ata o nivel 2: Dentro do nivel / e os seus primeiros cartafoles. Entón, tamén busca dentro de /tmp, co cal, busca nos ficheiros /tmp/users.csv e /tmp/Users.csv. A opción -H de grep permite no caso de atopar o patrón buscado imprimir o ficheiro no cal foi atopado.

```
$ find / -maxdepth 2 -type f -iname "u*.csv" -exec ls -l {} \; 2>/dev/null
```

Profundizando ata o nivel 2: Dentro do nivel / e os seus primeiros cartafoles. Entón, tamén busca dentro de /tmp, co cal, busca nos ficheiros /tmp/users.csv e /tmp/Users.csv. Pero, agora co buscado fai un listado extendido, é dicir, fai un listado extendido dos arquivos /tmp/users.csv e /tmp/Users.csv

```
$ find / -maxdepth 2 -type f -iname "u*.csv" -exec cp -pv {} /tmp/a1/ \; 2>/dev/null
```

Igual que o comando anterior pero agora fai unha copia dos arquivos /tmp/users.csv e /tmp/Users.csv ao cartafol /tmp/a1



```
$ find / -maxdepth 2 -type f -iname "u*.csv" 2>/dev/null
```

Busca ata o nivel 2. É dicir, en / e /tmp/ (e noutras rutas con niveis similares), co cal atopa os ficheiros /tmp/users.csv e /tmp/Users.csv pero non os que están contidos dentro de /tmp/a1

```
$ find / -maxdepth 3 -type f -iname "u*.csv" 2>/dev/null
```

Busca ata o nivel 3. É dicir, en / e /tmp/ e /tmp/a1 (e noutras rutas con niveis similares), co cal atopa os ficheiros /tmp/users.csv, /tmp/Users.csv, /tmp/a1/users.csv, /tmp/a1/Users.csv pero non os que están contidos dentro de /tmp/a1/a2

```
$ find / -maxdepth 4 -type f -iname "u*.csv" 2>/dev/null
```

Busca ata o nivel 4. É dicir, en / e /tmp/, /tmp/a1 e /tmp/a1/a2 (e noutras rutas con niveis similares), co cal atopa os ficheiros /tmp/users.csv, /tmp/Users.csv, /tmp/a1/users.csv, /tmp/a1/Users.csv e /tmp/a1/a2/users.csv

```
$ find / -maxdepth 4 -type f -iname "u*.csv" 2>/dev/null | xargs
```

Busca ata o nivel 4. É dicir, en / e /tmp/, /tmp/a1 e /tmp/a1/a2 (e noutras rutas con niveis similares), co cal atopa os ficheiros /tmp/users.csv, /tmp/Users.csv, /tmp/a1/users.csv, /tmp/a1/Users.csv e /tmp/a1/a2/users.csv. Mediante o xargs sen parámetros a saída en vez de saír nunha columna sae nunha fila.

```
$ find / -maxdepth 4 -type f -iname "u*.csv" 2>/dev/null | xargs -I {} ls -l {}
```

Busca ata o nivel 4. É dicir, en / e /tmp/, /tmp/a1 e /tmp/a1/a2 (e noutras rutas con niveis similares), co cal atopa os ficheiros /tmp/users.csv, /tmp/Users.csv, /tmp/a1/users.csv, /tmp/a1/Users.csv e /tmp/a1/a2/users.csv. Mediante o xargs co parámetro -I {} vaise recollendo cada valor atopado en {} para logo facerlle un listado extendido mediante ls -l

```
$ find / -maxdepth 4 -type f -iname "u*.csv" 2>/dev/null | xargs -I VAR ls -l VAR
```

Equivale ao comando anterior, soamente que agora a variable onde se recolle a búsqueda chámase VAR.

```
$ find / -maxdepth 4 -type f -iname "u*.csv" 2>/dev/null | xargs -I U ls -l U
```

Equivale ao comando anterior, soamente que agora a variable onde se recolle a búsqueda chámase U.

```
$ find / -type f -perm +4000 2>/dev/null | xargs -I U ls -l U
```

Atopar todos os arquivos do sistema que posúan o permiso SUID(4000) redireccionando calquera erro ao dispositivo /dev/null para logo facerlle un listado extendido mediante ls -l

```
$ find / -type f -perm +6000 2>/dev/null | xargs -I U ls -l U
```

Atopar todos os arquivos do sistema que posúan o permiso SUID(4000) ou GUID(2000) redireccionando calquera erro ao dispositivo /dev/null para logo facerlle un listado extendido mediante ls -l

```
$ find . -iname .* 2>/dev/null | xargs -I U ls -l U
```

Atopar profundizando dende a ruta actual calquera ficheiro/directorio que comece polo caracter punto, ou sexa, buscar os ficheiros/cartafolios ocultos do sistema redireccionando calquera erro ao dispositivo /dev/null para logo facerlle un listado extendido mediante ls -l

```
$ find . -iname .* 2>/dev/null | xargs -I U ls -l U
```

Equivale ao comando anterior.

- **mtime:** Marca temporal sobre a data e hora da última escritura nun ficheiro.  
Por exemplo, cando é modificado cun editor de textos.
- **atime:** Marca temporal sobre a última lectura dun ficheiro.  
Por exemplo, cando se accede a ler con cat, more...
- **ctime:** Marca temporal sobre o último cambio nun ficheiro en termos de inodo, é dicir, cambios en permisos, usuario...  
Por exemplo, cando é modificado con chown, chgrp, chmod...

```
$ find $HOME -mtime 0
```

Atopar en \$HOME os ficheiros que foron modificados nas últimas 24 horas.

```
$ find $HOME -mtime +5
```

Atopar en \$HOME os ficheiros que **non** foron modificados nos últimos 5 días.

```
$ find $HOME -mtime -5
```

Atopar en \$HOME os ficheiros que foron modificados nos últimos 5 días.

```
$ find $HOME -mtime +3 -mtime -5
```

Atopar en \$HOME os ficheiros que foron modificados entre 3 e 5 días.

```
$ find $HOME -atime -5
```

Atopar en \$HOME os ficheiros que foron leídos nos últimos 5 días.

```
$ find $HOME -ctime -5
```

Atopar en \$HOME os ficheiros que foron modificados en termos de inodo (permisos, usuario, grupo...) nos últimos 5 días.

```
$ find . -perm 664
```

Atopar profundizando dende a ruta actual calquera ficheiro con permisos 664(rw-rw-r--).

```
$ find . -size +10M
```

**Atopar profundizando dende a ruta actual calquera ficheiro con tamaño superior a 10M.**

```
$ find . -size -10M
```

**Atopar profundizando dende a ruta actual calquera ficheiro con tamaño inferior a 10M.**

```
$ find . -size 10M
```

**Atopar profundizando dende a ruta actual calquera ficheiro de tamaño 10M.**

```
$ find /etc/apache2 -user www-data
```

**Atopar profundizando dende a ruta /etc/apache2 calquera ficheiro que posúa como usuario propietario: www-data**

```
$ find /etc/apache2 -not -user www-data
```

**Atopar profundizando dende a ruta /etc/apache2 calquera ficheiro que non posúa como usuario propietario: www-data**

```
$ find /etc/apache2 ! -user www-data
```

**Equivale ao comando anterior.**

```
$ find /etc/apache2 ! -user www-data -o ! -group www-data
```

**Atopar profundizando dende a ruta /etc/apache2 calquera ficheiro que non posúa como usuario propietario ou(opción -o=or) grupo propietario: www-data e www-data respectivamente.**

```
$ find /etc/apache2 ! -user www-data -or ! -group www-data
```

**Equivale ao comando anterior: opción -o=or**

```
$ find /etc/apache2 -user www-data -a -group www-data
```

**Atopar profundizando dende a ruta /etc/apache2 calquera ficheiro que posúa como usuario propietario e(opción -a=and) grupo propietario: www-data e www-data respectivamente.**

```
$ find /etc/apache2 -user www-data -and -group www-data
```

**Equivale ao comando anterior: opción -a=and**

Ricardo Feijoo Costa



This work is licensed under a **Creative Commons Attribution-ShareAlike 4.0 International License**

# Comando sort (ordear)

## Opcións de interese

- **-f, --ignore-case**: Ordear ignorando maiúsculas/minúsculas.
- **-n, --numeric-host**: Ordear numericamente.
- **-r, --reverse**: Ordear de forma inversa, é dicir, invertindo a orde.
- **-k, --key=KEYDEF**: Ordear por columna.
- **-o, --output=FILE**: Enviar a saída do comando a un ficheiro.
- **-t, --field-separator=SEP**: Usar o campo que definamos como separador para ordear.
- **-u, --unique**: Amosa soamente a primeira coincidencia de varias.

## Práctica

O seguinte ficheiro, chamado **persoas.ods**:

	A	B	C	D
1	Nome	Apelido1	Apelido2	Idade
2	Luis	Suárez	Rodríguez	32
3	Ana	Pérez	López	44
4	Juana	Márquez	Otero	17
5	Braulio	Sánchez	Boullón	72
6	Pedro	Agrelo	Sousa	32
7	Carlos	Moreno	Peláez	67

foi gardado en formato CSV como **persoas.csv**:

```
Nome;Apelido1;Apelido2;Idade
Luis;Suárez;Rodríguez;32
Ana;Pérez;López;44
Juana;Márquez;Otero;17
Braulio;Sánchez;Boullón;72
Pedro;Agrelo;Sousa;32
Carlos;Moreno;Peláez;67
```

Tendo en conta este último ficheiro, **persoas.csv**, e a **separación co caracter punto e coma entre campos**:

```
$ sort persoas.csv #Por defecto sort ordear de forma alfanumérica cada liña según o primeiro carácter
```

```
Ana;Pérez;López;44
Braulio;Sánchez;Boullón;72
Carlos;Moreno;Peláez;67
Juana;Márquez;Otero;17
Luis;Suárez;Rodríguez;32
Nome;Apelido1;Apelido2;Idade
Pedro;Agrelo;Sousa;32
```

```
$ nl persoas.csv | sort #Por defecto sort ordear de forma alfanumérica cada liña según o primeiro carácter
```

```
1 Nome;Apelido1;Apelido2;Idade
2 Luis;Suárez;Rodríguez;32
3 Ana;Pérez;López;44
4 Juana;Márquez;Otero;17
5 Braulio;Sánchez;Boullón;72
6 Pedro;Agrelo;Sousa;32
7 Carlos;Moreno;Peláez;67
```

```
$ nl persoas.csv | sort -r #Por defecto sort ordear de forma alfanumérica cada liña según o primeiro carácter e coa opción -r amosa a saída de forma inversa.
```

```
7 Carlos;Moreno;Peláez;67
6 Pedro;Agrelo;Sousa;32
5 Braulio;Sánchez;Boullón;72
4 Juana;Márquez;Otero;17
3 Ana;Pérez;López;44
2 Luis;Suárez;Rodríguez;32
1 Nome;Apelido1;Apelido2;Idade
```

```
$ sort -r persoas.csv #Por defecto sort ordear de forma alfanumérica cada liña según o primeiro carácter e coa opción -r amosa a saída de forma inversa.
```

```
Pedro;Agrelo;Sousa;32
Nome;Apelido1;Apelido2;Idade
Luis;Suárez;Rodríguez;32
Juana;Márquez;Otero;17
Carlos;Moreno;Peláez;67
Braulio;Sánchez;Boullón;72
Ana;Pérez;López;44
```

\$ sort -r -o file.csv pessoas.csv #Por defecto sort ordeas de forma alfanumérica cada liña según o primeiro carácter e coa opción **-r** amosa a saída de forma inversa. Coa opción **-o** envía a saída ao ficheiro file.csv. A opción **-o** é similar á opción de redirección >

```
$ cat file.csv
Pedro;Agreló;Sousa;32
Nome;Apelido1;Apelido2;Idade
Luis;Suárez;Rodríguez;32
Juana;Márquez;Otero;17
Carlos;Moreno;Peláez;67
Braulio;Sánchez;Boullón;72
Ana;Pérez;López;44
```

\$ sort -t ';' -k1 pessoas.csv #Ordear de forma alfanumérica según a columna 1, interpretando cada columna a parte do ficheiro separada polo carácter punto e coma mediante a opción **-t**.

```
Ana;Pérez;López;44
Braulio;Sánchez;Boullón;72
Carlos;Moreno;Peláez;67
Juana;Márquez;Otero;17
Luis;Suárez;Rodríguez;32
Nome;Apelido1;Apelido2;Idade
Pedro;Agreló;Sousa;32
```

\$ sort -t ';' -k2 pessoas.csv #Ordear de forma alfanumérica según a columna 2, interpretando cada columna a parte do ficheiro separada polo carácter punto e coma mediante a opción **-t**.

```
Pedro;Agreló;Sousa;32
Nome;Apelido1;Apelido2;Idade
Juana;Márquez;Otero;17
Carlos;Moreno;Peláez;67
Ana;Pérez;López;44
Braulio;Sánchez;Boullón;72
Luis;Suárez;Rodríguez;32
```

\$ sort -t ';' -k3 pessoas.csv #Ordear de forma alfanumérica según a columna 3, interpretando cada columna a parte do ficheiro separada polo carácter punto e coma mediante a opción **-t**.

```
Nome;Apelido1;Apelido2;Idade
Braulio;Sánchez;Boullón;72
Ana;Pérez;López;44
Juana;Márquez;Otero;17
Carlos;Moreno;Peláez;67
Luis;Suárez;Rodríguez;32
Pedro;Agreló;Sousa;32
```

\$ sort -t ';' -k4 pessoas.csv #Ordear de forma alfanumérica según a columna 4, interpretando cada columna a parte do ficheiro separada polo carácter punto e coma mediante a opción **-t**.

```
Juana;Márquez;Otero;17
Luis;Suárez;Rodríguez;32
Pedro;Agreló;Sousa;32
Ana;Pérez;López;44
Carlos;Moreno;Peláez;67
Braulio;Sánchez;Boullón;72
Nome;Apelido1;Apelido2;Idade
```

\$ sort -t ';' -k4 -u pessoas.csv #Ordear de forma alfanumérica según a columna 4, interpretando cada columna a parte do ficheiro separada polo carácter punto e coma mediante a opción **-t**. Coa opción **-u** soamente aparecerá a primeira aparición de varias concurrencias que sexan iguais. Así, nesta caso soamente aparecerá unha soa liña co número 32

```
Juana;Márquez;Otero;17
Luis;Suárez;Rodríguez;32
Ana;Pérez;López;44
Carlos;Moreno;Peláez;67
Braulio;Sánchez;Boullón;72
Nome;Apelido1;Apelido2;Idade
```

\$ sort -t ';' -k1 pessoas.csv | grep -v 'Nome' #Ordear de forma alfanumérica según a columna 1, interpretando cada columna a parte do ficheiro separada polo carácter punto e coma mediante a opción **-t**. Ademais non amosar a cabeceira (primeira liña) do ficheiro pessoas.csv

```
Ana;Pérez;López;44
Braulio;Sánchez;Boullón;72
Carlos;Moreno;Peláez;67
Juana;Márquez;Otero;17
Luis;Suárez;Rodríguez;32
Pedro;Agreló;Sousa;32
```

\$ tail -n +2 persoas.csv | sort -t ';' -k1 #Ordear de forma alfanumérica según a columna 1, interpretando cada columna a parte do ficheiro separada polo caracter punto e coma mediante a opción -t. Ademais non amosar a cabeceira (primeira liña) do ficheiro persoas.csv

```
Ana;Pérez;López;44
Braulio;Sánchez;Boullón;72
Carlos;Moreno;Peláez;67
Juana;Márquez;Otero;17
Luis;Suárez;Rodríguez;32
Pedro;Agrelo;Sousa;32
```

\$ sort -t ';' -k4 -nr persoas.csv | grep -v 'Nome' #Ordear de forma alfanumérica según a columna 4 (Idade) de maior a menor, interpretando cada columna a parte do ficheiro separada polo caracter punto e coma mediante a opción -t. Ademais non amosar a cabeceira (primeira liña) do ficheiro persoas.csv

```
Braulio;Sánchez;Boullón;72
Carlos;Moreno;Peláez;67
Ana;Pérez;López;44
Pedro;Agrelo;Sousa;32
Luis;Suárez;Rodríguez;32
Juana;Márquez;Otero;17
```

\$ tail -n +2 persoas.csv | sort -t ';' -k4 -nr #Ordear de forma alfanumérica según a columna 4 (Idade) de maior a menor, interpretando cada columna a parte do ficheiro separada polo caracter punto e coma mediante a opción -t. Ademais non amosar a cabeceira (primeira liña) do ficheiro persoas.csv

```
Braulio;Sánchez;Boullón;72
Carlos;Moreno;Peláez;67
Ana;Pérez;López;44
Pedro;Agrelo;Sousa;32
Luis;Suárez;Rodríguez;32
Juana;Márquez;Otero;17
```

\$ sort -t ';' -k4 -nr persoas.csv | grep -v 'Nome' | head -2 #Ordear de forma alfanumérica según a columna 4 (Idade) de maior a menor, interpretando cada columna a parte do ficheiro separada polo caracter punto e coma mediante a opción -t. Ademais non amosar a cabeceira (primeira liña) do ficheiro persoas.csv, e amosar soamente as 2 primeiras liñas

```
Braulio;Sánchez;Boullón;72
Carlos;Moreno;Peláez;67
```

\$ sort -t ';' -k4 -nr persoas.csv | grep -v 'Nome' | head -2 | sort -t ';' -k2 #Ordear de forma alfanumérica según a columna 4 (Idade) de maior a menor, interpretando cada columna a parte do ficheiro separada polo caracter punto e coma mediante a opción -t. Ademais non amosar a cabeceira (primeira liña) do ficheiro persoas.csv, amosar soamente as 2 primeiras liñas e ordealas alfanumericamente pola columna 2 (caracter separador punto e coma)

```
Carlos;Moreno;Peláez;67
Braulio;Sánchez;Boullón;72
```

Ricardo Feijoo Costa



This work is licensed under a [Creative Commons Attribution-ShareAlike 4.0 International License](https://creativecommons.org/licenses/by-sa/4.0/)

# (Des)Empaquetar e (Des)Comprimir

## tar, gzip, gunzip, zcat, bzip2, bunzip2, zip, unzip

- tar: Empaqueta ficheiros e directorios/(Des)Comprime ficheiros. Extensões típicas: .tar, .tar.gz, .tgz, .tar.bz2
- gzip: (Des)Comprime ficheiros e directorios. Extensão típica: .gz
- gunzip: Descomprime ficheiros e directorios. Extensões típicas: .gz, .tar.gz, .tgz
- zcat: Descomprime ficheiros e directorios. Idéntico a gunzip -c ; Extensões típicas: As equivalentes a gunzip.
- bzip2: (Des)Comprime ficheiros. Extensão típica: .bz2
- bunzip2: Descomprime ficheiros. Extensão típica: .bz2
- bzcat: Descomprime ficheiros e directorios. Idéntico a bunzip2 -c ; Extensões típicas: As equivalentes a bunzip2.
- zip: Comprime ficheiros. Extensão típica: .zip
- unzip: Descomprime ficheiros. Extensão típica: .zip

## tar

- tar cvf : Empaquetar
- tar tvf: Ver contido do empaquetado
- tar xvf: Desempaquetar
- tar cvfz : Comprimir en tar.gz
- tar tvfz: Ver contido do comprimido en tar.gz
- tar xvfz: Descomprimir en tar.gz
- tar cvfj : Comprimir en tar.bz2
- tar tvfj: Ver contido do comprimido en tar.bz2
- tar xvfj: Descomprimir en tar.bz2

```
$ mkdir /tmp/pack #Criar o directorio /tmp/pack
```

```
$ cd /tmp/pack #Acceder ao directorio /tmp/pack
```

```
$ tar cvf file1.tar /etc/passwd /etc/group #Criar un ficheiro empaquetado file1.tar co contido no directorio etc e dentro deste os ficheiros passwd e group, é dicir, empaqueta etc/passwd e etc/group
```

```
$ tar tvf file1.tar #Ver o contido do ficheiro empaquetado file1.tar
```

```
$ tar xvf file1.tar #Desempaquetar o contido do ficheiro file1.tar. Así, crearíase a seguinte estrutura dentro de /tmp/pack:
```

```
etc/
├── group
└── passwd
```

```
0 directories, 2 files
```

```
$ rm -rf etc/ #Eliminar de forma recursiva e sen preguntar o cartafol /tmp/pack/etc
```

```
$ tar xvf file1.tar etc/passwd #Desempaquetar soamente a estrutura etc/passwd:
```

```
etc
└── passwd
```

```
0 directories, 1 file
```

```
$ rm -rf etc/ #Eliminar de forma recursiva e sen preguntar o cartafol /tmp/pack/etc
```

```
$ tar Pcvf file2.tar /etc/passwd /etc/group #Criar un ficheiro empaquetado file2.tar co contido do directorio /etc e dentro deste os ficheiros passwd e group sen eliminar a barra '/' inicial dos nomes, é dicir, empaqueta /etc/passwd, /etc/group e non etc/passwd, etc/group
```

```
$ tar Ptvf file2.tar #Ver o contido do ficheiro empaquetado file2.tar sen eliminar a barra '/' inicial dos nomes
```



\$ tar Pxvf file2.tar #Desempaquetar o contido do ficheiro file2.tar, pero como o ficheiro foi empaquetado coa opción -P non se eliminaron as barras '/', co cal agora os ficheiros serán desempaquetados no /, é dicir, en /etc/passwd e /etc/group, polo que non se van poden desempaquetar debido a que a execución estase a facer cun usuario sen permisos de escritura neses ficheiros.

\$ cd /tmp #Acceder ao directorio /tmp

\$ tar cvf file3.tar /tmp/pack /etc/passwd /etc/group #Crear un ficheiro empaquetado file3.tar co contido do directorio /tmp/pack e os ficheiros passwd e group eliminando as barras iniciais '/' dos nomes, é dicir, empaqueta tmp/pack, etc/passwd, etc/group e non /tmp/pack, /etc/passwd, /etc/group

\$ tar cvfz file4.tar.gz /tmp/pack /etc/passwd /etc/group #Crear un ficheiro comprimido mediante gzip do que foi empaquetado mediante tar, o cal contén o contido do directorio /tmp/pack e os ficheiros passwd e group eliminando as barras iniciais '/' dos nomes, é dicir, empaqueta tmp/pack, etc/passwd, etc/group e non /tmp/pack, /etc/passwd, /etc/group

\$ tar cvfj file5.tar.bz2 /tmp/pack /etc/passwd /etc/group #Crear un ficheiro comprimido mediante bzip2 do que foi empaquetado mediante tar, o cal contén o contido do directorio /tmp/pack e os ficheiros passwd e group eliminando as barras iniciais '/' dos nomes, é dicir, empaqueta tmp/pack, etc/passwd, etc/group e non /tmp/pack, /etc/passwd, /etc/group

\$ tar tvfz file4.tar.gz #Ver o contido do ficheiro empaquetado e comprimido file4.tar.gz

\$ tar tvfj file5.tar.bz2 #Ver o contido do ficheiro empaquetado e comprimido file5.tar.bz2

\$ tar xvfz file4.tar.gz #Descomprimir o contido do ficheiro empaquetado e comprimido file4.tar.gz

\$ tar xvfj file5.tar.bz2 #Descomprimir o contido do ficheiro empaquetado e comprimido file5.tar.bz2

```
$ mkdir /tmp/comprimir #Crear o directorio /tmp/comprimir
```

```
$ cd /tmp/comprimir #Acceder ao directorio /tmp/comprimir
```

```
$ cp -pv /etc/passwd /etc/group /etc/motd /etc/issue . #Copiar preservando os permisos e marcas  
temporais os ficheiros /etc/passwd, /etc/group, /etc/motd e /etc/issue na ruta actual, ou sexa, dentro de  
/tmp/comprimir
```

### gzip, gunzip, zcat: (Des)comprimir ficheiros gz

- gzip : Comprimir. Niveis de compresión entre o 1(peor) e o 9(mellor).
- gzip -d : Descomprimir
- gzip -c : Comprimir amosando na consola(saída estándar) o contido comprimido, pero preservando o ficheiro descomprimido. Non crea ficheiro .gz
- gzip -dc : Descomprimir amosando na consola(saída estándar) o contido descomprimido, pero preservando o ficheiro comprimido. Non se restaura o ficheiro orixinal.
- gzip -l : Listar información de compresión
  
- gunzip : Descomprimir. Equivale a gzip -d
- gunzip -d : Equivale a gunzip
- gunzip -c : Equivale a gzip -dc
- gunzip -dc : Equivale a gunzip -c
- gunzip -l : Equivale a gzip -l
  
- zcat: Equivale a gunzip -c

```
$ gzip passwd group #Comprimir passwd e group, creando passwd.gz e group.gz  
respectivamente, sen preservar os ficheiros sen comprimir
```

```
$ gzip -1 motd #Comprimir co peor nivel de compresión o ficheiro motd, creando motd.gz, sen  
preservar o ficheiro motd
```

```
$ gzip -9 issue #Comprimir co mellor nivel de compresión o ficheiro issue, creando issue.gz, sen  
preservar o ficheiro issue
```

```
$ gzip -l motd.gz issue.gz #Listar información de compresión dos ficheiros motd.gz e issue.gz
```

```
$ gzip -lv motd.gz issue.gz #Listar información de compresión en modo detallado (verbose) dos  
ficheiros motd.gz e issue.gz
```

```
$ gzip -dc motd.gz #Descomprimir amosando na consola(saída estándar) o contido  
descomprimido de motd.gz, pero preservando o ficheiro comprimido motd.gz. Non se restaura o  
ficheiro orixinal.
```

```
$ gzip -d motd.gz #Descomprimir o ficheiro motd.gz, restaurando o ficheiro orixinal motd.
```

```
$ gunzip -c issue.gz #Descomprimir amosando na consola(saída estándar) o contido  
descomprimido de issue.gz, pero preservando o ficheiro comprimido issue.gz. Non se restaura o  
ficheiro orixinal.
```

```
$ gunzip issue.gz #Descomprimir o ficheiro issue.gz, restaurando o ficheiro orixinal issue.
```

```
$ gunzip -l passwd.gz group.gz #Listar información de compresión dos ficheiros passwd.gz e  
group.gz
```

```
$ zcat group.gz: #Equivale a gunzip -c group.gz, entón equivale a descomprimir amosando na  
consola(saída estándar) o contido descomprimido de group.gz, pero preservando o ficheiro  
comprimido group.gz. Non se restaura o ficheiro orixinal.
```

## bzip2, bunzip2, bzipcat: (Des)comprimir ficheiros bz2

- bzip2 : Comprimir. Niveis de compresión entre o 1(peor) e o 9(mellor).
- bzip2 -d : Descomprimir
- bzip2 -c : Comprimir amosando na consola(saída estándar) o contido comprimido, pero preservando o ficheiro descomprimido. Non crea ficheiro .bz2
- bzip2 -dc : Descomprimir amosando na consola(saída estándar) o contido descomprimido, pero preservando o ficheiro comprimido. Non se restaura o ficheiro orixinal.
  
- bunzip2 : Descomprimir. Equivale a bzip2 -d
- bunzip2 -d : Equivale a bunzip2
- bunzip2 -c : Equivale a bzip2 -dc
- bunzip2 -dc : Equivale a bunzip2 -c
  
- bzipcat: Equivale a bunzip2 -c

\$ cp -pv /etc/passwd /etc/group . #Copiar preservando os permisos e marcas temporais os ficheiros /etc/passwd e /etc/group na ruta actual, ou sexa, dentro de /tmp/comprimir

\$ bzip2 passwd group #Comprimir passwd e group, creando passwd.bz2 e group.bz2 respectivamente, sen preservar os ficheiros sen comprimir

\$ bzip2 -1 motd #Comprimir co peor nivel de compresión o ficheiro motd, creando motd.bz2, sen preservar o ficheiro motd

\$ bzip2 -9 issue #Comprimir co mellor nivel de compresión o ficheiro issue, creando issue.bz2, sen preservar o ficheiro issue

\$ bzip2 -dc motd.bz2 #Descomprimir amosando na consola(saída estándar) o contido descomprimido de motd.bz2, pero preservando o ficheiro comprimido motd.bz2. Non se restaura o ficheiro orixinal.

\$ bzip2 -d motd.bz2 #Descomprimir o ficheiro motd.bz2, restaurando o ficheiro orixinal motd.

\$ bunzip2 -c issue.bz2 #Descomprimir amosando na consola(saída estándar) o contido descomprimido de issue.bz2, pero preservando o ficheiro comprimido issue.bz2. Non se restaura o ficheiro orixinal.

\$ bunzip2 issue.bz2 #Descomprimir o ficheiro issue.bz2, restaurando o ficheiro orixinal issue.

\$ bzipcat group.bz2: #Equivale a bunzip2 -c group.bz2, entón equivale a descomprimir amosando na consola(saída estándar) o contido descomprimido de group.bz2, pero preservando o ficheiro comprimido group.bz2. Non se restaura o ficheiro orixinal.

## zip, unzip

- zip : Comprimir
- zip -r : Comprimir recursivamente, é dicir, toda a estrutura arbórea pertencente a un directorio
- unzip : Descomprimir.
- unzip -c : Descomprimir amosando na consola(saída estándar) o contido comprimido, preservando o ficheiro comprimido.
- unzip -l : Listar o contido do comprimido
- unzip -d dir : Descomprimir dentro do directorio dir. Se non existe o directorio créase.

\$ zip issue.zip /etc/issue #Comprimir o ficheiro /etc/issue, creando o ficheiro issue.zip na ruta actual

\$ zip -r comprimir.zip /tmp/comprimir #Comprimir todo o contido do directorio /tmp/comprimir (ficheiros, directorios e subdirectorios) creando o ficheiro comprimir.zip na ruta actual

\$ unzip comprimir.zip #Descomprimir o ficheiro comprimir.zip, preservando o ficheiro comprimido comprimir.zip e restaurando na ruta actual en tmp (./tmp/comprimir) a estrutura orixinal.

\$ unzip -c issue.zip #Descomprimir amosando na consola(saída estándar) o contido descomprimido de issue.zip, preservando o ficheiro comprimido issue.zip e non se restaura o ficheiro orixinal.

\$ unzip -l issue.zip #Listar o contido do ficheiro comprimido issue.zip

Ricardo Feijoo Costa



This work is licensed under a **Creative Commons Attribution-ShareAlike 4.0 International License**

# Xestión de procesos:

ps, pstree, pgrep, pidof, kill, killall, pkill, top, uptime, &, jobs, fg, <Ctrl>+<z>, <Ctrl>+<c>, bg, nohup, disown, nice, renice, lsof, fuser

- Abrir unha consola, sen permisos de **root**, e executar o comando:

```
$ yes > /dev/null &
$ yes
```

E agora qué? Cómo recuperar a consola? De momento <Ctrl>+<c>

## Práctica

- ps: amosa a lista de procesos que se están executando.  
\$ ps aux  
\$ ps aux | grep yes  
\$ ps -ef | grep yes  
\$ ps -efjH
- pstree: ps en forma de árbore.  
\$ pstree  
\$ pstree -paA  
\$ pstree -paA | grep yes
- pgrep: ps + grep. Localiza o PID a través do nome.  
\$ pgrep -u root init  
\$ pgrep -u root systemd  
\$ pgrep yes
- pidof: amosa o PID do proceso que solicitamos.  
\$ pidof init  
\$ pidof systemd  
\$ pidof yes
- kill: envía un sinal(número ou nome, ver kill -l) a un proceso (PID).  
\$ kill -l

**man 7 signal (ver táboa explicativa procesos)**

O sinal poder ser un número ou un nome, por exemplo: **kill -9** equivale a **kill -s SIGKILL**, que tamén equivale a **kill -SIGKILL**.  
Sinais moi empregados:

- **1) SIGHUP**: O proceso pai envía o sinal HANG UP a todos os seus procesos fillos cando remata
- **2) SIGINT**: <Ctrl>+<C>
- **3) SIGQUIT**: <Ctrl>+<C> + "file coredump"
- **9) SIGKILL**: kill inmediato forzoso ao proceso, sen importar o estado do/s proceso/s pai/s ou herdados, co cal pode deixar procesos sen rematar as peticións pendentes e outros sen eliminar os PID.
- **15) SIGTERM**: Por defecto. Remate do proceso limpo
- **18) SIGCONT**: Continúa se o proceso foi suspendido.
- **19) SIGSTOP**: Suspende un proceso

```
$ kill -9 `pidof yes`
$ pidof yes
$ yes > /dev/null &
$ kill -19 $(pidof yes)
```

```
[1]+  Detenido          yes > /dev/null
```

```
$ kill -18 $(pidof yes)
$ jobs #Amosa as tarefas da sesión actual en segundo plano.
```

```
[1]+  Ejecutando          yes > /dev/null &
```

```
$ kill -9 $(pidof yes)
$ pidof yes
```

- killall: envía un sinal a un proceso (nome).  
\$ yes > /dev/null &  
\$ killall yes  
\$ pidof yes  
\$ yes > /dev/null &
- pkill: ps + kill. Localiza o PID a través do nome.  
\$ pkill yes  
\$ yes > /dev/null &
- top: amosa información de procesador/procesos/memoria: h,q,?,<Esc>,→,←,↑,↓,<AvPág>,<RePág>,1,L,R,c,u,k,r,q
  - ¿Cantos recursos consume o proceso **yes** ?
- uptime: indica a hora actual, o tempo que o sistema está en marcha, o número de usuarios conectados e a carga promedio do sistema para os últimos 1, 5 e 15 minutos.  
\$ uptime

- &: background; ao lanzar un proceso lánzao en segundo plano.  
\$ yes > /dev/null &  
\$ yes > /dev/null &  
\$ yes > /dev/null &
- jobs: amosa as tarefas da sesión actual en segundo plano.  
\$ jobs  
[1] Ejecutando yes > /dev/null &  
[2] Ejecutando yes > /dev/null &  
[3]- Ejecutando yes > /dev/null &  
[4]+ Ejecutando yes > /dev/null &
- fg: leva un proceso a executarse en primeiro plano.  
\$ fg %3  
yes > /dev/null
- <Ctrl>+<z>: interrompe un proceso e suspéndeo.  
\$ <Ctrl>+<z>  
[3]+ Detenido yes > /dev/null  
\$ jobs  
[1] Ejecutando yes > /dev/null &  
[2] Ejecutando yes > /dev/null &  
[3]+ Detenido yes > /dev/null  
[4]- Ejecutando yes > /dev/null &
- bg: manda un proceso a executarse en segundo plano.  
\$ bg %3  
[3]+ yes > /dev/null &  
\$ jobs  
[1] Ejecutando yes > /dev/null &  
[2] Ejecutando yes > /dev/null &  
[3]- Ejecutando yes > /dev/null &  
[4]+ Ejecutando yes > /dev/null &
- <Ctrl>+<c>: interrompe un proceso e párao (sinal KILL).  
\$ fg %4  
\$ Ctrl+c  
\$ jobs  
[1] Ejecutando yes > /dev/null &  
[2]- Ejecutando yes > /dev/null &  
[3]+ Ejecutando yes > /dev/null &
- nohup: fai que un proceso colgue de init(systemd)(proceso 1), evita que sexa alterado polo sinal SIGHUP(1) e redirixe a súa saída ao arquivo *nohup.out*.  
\$ mousepad &  
Pechar o terminal no botón aspa da opción gráfica. Vemos que se pecha o mousepad. Agora abrir un terminal.  
\$ nohup mousepad &  
Pechar o terminal no botón aspa da opción gráfica. Agora vemos que non se pecha o mousepad. Abrir de novo un terminal.
- disown: fai un nohup dun proceso xa en execución.  
\$ mousepad  
\$ <Ctrl>+<z>  
\$ bg %1  
\$ disown -h %1  
Pechar o terminal no botón aspa da opción gráfica. Agora vemos que non se pecha o mousepad. Abrir de novo un terminal.
- nice: establece unha prioridade ao proceso comprendida no intervalo [-20,19], sendo -20 o máis favorable e o 19 o menos favorable. Por defecto cada proceso toma valor cero. Soamente os usuarios con permisos de root poden establecer prioridade negativa.  
\$ nice -n -20 cp -pv /etc/passwd /tmp/  
\$ nice -n 19 ls /tmp  
# nice -n -20 cp -pv /etc/passwd /tmp/  
# nice -n -5 ls /tmp



- renice: reestablece a prioridade dun proceso.  
\$ yes > /dev/null &  
\$ renice -n 19 -p PID  
\$ renice -n -20 -u root
- lsof, fuser: permite visualizar os ficheiros e sockets en uso polas aplicacións.  
\$ lsof -i  
\$ su - -c "lsof -i"  
# lsof -i  
# lsof +D /var/log  
# lsof -c init  
# lsof -c systemd  
# lsof /  
# lsof -u username  
# lsof -p PID  
# lsof -i -a -p PID  
# fuser /dev/tty1  
# fuser .  
# fuser -v .  
# fuser --mount mountname  
# fuser -v -n tcp PORT  
# fuser -km mountname

Ricardo Feijoo Costa



This work is licensed under a [Creative Commons Attribution-ShareAlike 4.0 International License](https://creativecommons.org/licenses/by-sa/4.0/)

# Xestión de usuarios:

**su, (/etc/sudoers) sudo e visudo, who, w, last, lastb, lastlog, /etc/passwd, /etc/shadow, /etc/group, /etc/gshadow, id, whoami, groups, useradd, passwd, usermod, chsh, userdel, groupadd, gpasswd, groupdel, groupmod, chage, chfn**

- su: Permite abrir unha sesión(console) co ID (identificador) doutro usuario ou iniciar unha nova shell con ese identificador.  
\$ su username **#Abrir unha (sub)console para o usuario username.**  
\$ su **#Abrir unha (sub)console para o usuario root.**  
\$ su - username **#Abrir unha (sub)console para o usuario username cargando as súas variables de entorno. A nova console equivale á que tería o usuario cando fai login.**  
\$ su -c "comando" **#Executar o comando sendo root.**  
\$ su - username -c "comando" **#Executar o comando sendo o usuario username.**
- sudo: Executar os comandos permitidos en nome doutros usuarios. Permisos en /etc/sudoers. Editar este ficheiro con visudo.  
\$ sudo su - **#Acceder como root como se fixera login sen sudo. Comúmente empregado nas iso live de GNU/Linux para convertirse en root.**  
\$ sudo umount /dev/sdd1 **#Desmontar a unidade /dev/sdd1, optendo permisos a través de sudo.**

- visudo: Editar /etc/sudoers: Cmnd\_Alias (Alias de comandos), User\_Alias (Alias de usuarios ou grupos), Run\_Alias (Igual que User\_Alias podendo empregar #uid), e Host\_Alias (Alias de Hosts). Regras de acceso

## **/etc/sudoers: Alias**

**Cmnd\_Alias NOMEGRUPO = comando1, comando2, ..., comandoN #Comandos que se poden empregar con sudo se alguén emprega o alias definido como NOMEGRUPO.**

- **Cmnd\_Alias SERVICIOS = /etc/init.d/ssh restart, /etc/init.d/ssh reload #O alias SERVICIOS identifica os comandos: /etc/init.d/ssh restart, /etc/init.d/ssh reload. Entón, a quen se lle aplique este alias poderá executar eses comandos.**

**User\_Alias NOMEGRUPO= [user|group]name1, [user|group]name2, ..., [user|group]nameN  
#Usuarios/grupos que poden empregar con sudo**

- **User\_Alias ADMINS = pepito, fulanito, %lpadmin, !manganito #Identifica a quen se lle aplica o alias ADMINS, a saber: si se lle aplica aos usuarios pepito, fulanito e ao grupo lpadmin, pero non se lle pode aplicar ao usuario manganito**

**Runas\_Alias NOMEGRUPO= [user[#uid\_user]]groupname1, [user[#uid\_user]]groupname2, ..., [user[#uid\_user]]groupnameN #Usuarios/grupos que poden empregar con sudo**

- **Runas\_Alias ADMINS = #uid\_pepito, fulanito, %lpadmin, !manganito #Identifica a quen se lle aplica o alias ADMINS, a saber: si se lle aplica aos usuarios pepito, fulanito e ao grupo lpadmin, pero non se lle pode aplicar ao usuario manganito**

**Host\_Alias NOMEGRUPO= [IP|IP/MS|NET/MS|HOSTNAME|DNS] #Hosts**

- **Host\_Alias LAN = 192.168.100.10, webserver #O alias LAN aplícaselle á IP 192.168.100.10 e ao host de nome webserver**

## **/etc/sudoers: Regras de acceso**

**[usuario|%grupo, Alias] [host] = [comando1, comando2, comandoN | NOMEGRUPO]  
onde:**

**host, pode ser: ALL, IP, IP/MS, NET/MS, Host\_Alias**

- **pepito ALL = SERVICIOS #O usuario pepito pode executar en calquera host o Alias SERVICIOS**
- **ADMINS ALL = /sbin/ifconfig #Os usuarios definidos no alias ADMIN en calquera host poden executar o comando /sbin/ifconfig**
- **pepito ALL = (ALL) ALL #O usuario pepito pode executar en calquera host calquera comando de calquera usuario incluído os de root**
- **pepito ALL = (root) NOPASSWD: ALL #O usuario pepito pode executar en calquera host calquera comando de root e non se lle pedirá contrasinal**

- who: Amosa quen está "logueado". Logs de /var/run/utmp  
\$ who  
\$ who -a
- w: Amosa quen está "logueado" e que está facendo. Logs de /var/run/utmp  
\$ w  
\$ w -l usuario
- last(logs de /var/log/wtmp), lastb (logs de /var/log/btmp), lastlog(logs de /var/log/lastlog): Información sobre accesos dos usuarios ao sistema.  
\$ last  
\$ lastb  
\$ exit; su - **#Trabucarse co contrasinal --> por ben o contrasinal na segunda petición do mesmo.**  
\$ lastb  
\$ lastlog

- uptime: información sobre o tempo que o sistema leva encendido, o número de usuarios conectados e a carga media do sistema.  
\$ uptime
- /etc/passwd,/etc/shadow,etc/group,/etc/gshadow: Repectivamente ficheiros: información usuarios, contrasinais usuarios, grupos, contrasinais grupos
- id: Amosa información de usuario e grupo dun usuario.  
\$ id  
\$ id -un  
\$ id username  
\$ id -un username
- whoami: Amosa o usuario co que se están a executar os comandos. Equivale a id -un.  
\$ whoami
- groups: Amosa os grupos aos cales pertence un usuario.  
\$ groups #Amosa os grupos aos cales pertence o usuario que executa o comando  
\$ groups username #Amosa os grupos aos cales pertence o usuario de nome username  
\$ groups root #Amosa os grupos aos cales pertence o usuario root
- useradd: Crear usuario.  
# useradd -m -d /home/username -s /bin/bash username #Xerar usuario con:  
1. -m: Crea \$HOME coa estrutura de ficheiros según modelo /etc/skel  
2. -d: Cartafol \$HOME=/home/username  
3. -s: Shell \$BASH=/bin/bash
- passwd: Xerar/modificar contrasinal  
\$ passwd #Modificar contrasinal do usuario que executa o comando  
# passwd username #Modificar contrasinal do usuario username  
# passwd -l username #Bloquea contrasinal do usuario username  
# passwd -u username #Desbloquea contrasinal do usuario username
- usermod: Modificar usuario  
# usermod -g vboxdrv username #Forza o grupo vboxdrv para o grupo do usuario username  
# usermod -a -G vboxdrv username #Engadir aos grupos secundarios o grupo vboxdrv para o usuario username  
# usermod -L username #Bloquea a conta do usuario username  
# usermod -U username #Desbloquea a conta do usuario username  
# usermod -s /usr/sbin/nologin username #Cambia a shell do usuario username a /usr/sbin/nologin: Acceso mediante sudo ou ftp  
# usermod -s /bin/false username #Cambia a shell do usuario username a /bin/false: Sen acceso.
- chsh: Cambia a shell de traballo do usuario  
# chsh -s /usr/sbin/nologin username #Cambia a shell do usuario username a /usr/sbin/nologin: Acceso mediante sudo ou ftp  
# chsh -s /bin/false username #Cambia a shell do usuario username a /bin/false: Sen acceso.
- userdel: Eliminar usuario  
# userdel username #Eliminar o usuario username, pero segue existindo o seu cartafol \$HOME co seu contido.  
# userdel -r username #Eliminar o usuario username, e tamén o seu cartafol \$HOME.
- groupadd: Crear grupo  
# groupadd newgroup #Crea o grupo newgroup.  
# groupadd -g 10101 newgroup #Crea o grupo newgroup co seu gid=10101
- gpasswd: Crear/modificar contrasinal de grupo  
# gpasswd newgroup #Crea/modifica contrasinal para o grupo newgroup.
- groupdel: Eliminar grupo  
# groupdel newgroup #Elimina o grupo newgroup.
- groupmod: Modificar grupo  
# groupmod -n GROUPNEW newgroup #Cambia o nome do grupo newgroup a GROUPNEW.
- chage: Cambiar tempo de expiración dun contrasinal de usuario: último cambio de contrasinal, caducidade do contrasinal, contrasinal inactiva, caducidade da conta do usuario, número de días mínimo e máximo entre cambios de contrasinal, número de días de aviso antes de caducidade do contrasinal  
\$ chage -h #Amosa axuda sobre o comando chage.  
\$ chage -l username #Amosa información sobre as datas de caducidade dos contrasinais do usuario de nome username.  
\$ chage username #Sen opcións o comando chage opera de forma interactiva, preguntando os parámetros de caducidade, neste caso para o usuario de nome username.  
\$ chage -E 2020-02-01 username #Establece a data de caducidade do contrasinal do usuario de nome username ao 1 de febreiro de 2020.  
\$ chage -l 2 username #Deshabilita a conta do usuario de nome username logo de 2 días da data de caducidade do contrasinal.  
\$ chage -m 10 username #Cambia a 10 o número mínimo de días antes de cambiar o contrasinal para o usuario de nome username.  
\$ chage -M 20 username #Cambia a 20 o número máximo de días antes de cambiar o contrasinal para o usuario de nome username.

- `chfn`: Modificar campo gecol (comentario sobre a información do usuario)
  - # `chfn` #Sen opcións o comando `chfn` opera de forma interactiva, preguntando os parámetros gecol para o usuario que executa o comando, a saber: nome completo(fullname), número de habitación(dirección), teléfono do traballo, teléfono da casa e outros. Por defecto o usuario sen permisos de root non pode modificar nin o seu nome completo (fullname), nin outros. Os parámetros gecol aparecen separados por caracteres coma ',' no ficheiro `/etc/passwd`
  - # `chfn username` #Cambiar o campo gecol para o usuario de nome username. Soamente root pode modificar o campo gecol doutro usuario.
  - # `chfn -f 'Nome Apelido1 Apelido2' username` #Cambiar o fullname do campo gecol para o usuario de nome username. Soamente root pode modificar o fullname dun usuario.

Ricardo Feijoo Costa



This work is licensed under a [Creative Commons Attribution-ShareAlike 4.0 International License](https://creativecommons.org/licenses/by-sa/4.0/)

# Repositorios

## /etc/apt/sources.list

```
deb protocol://site.example.com/debian distribution component1 component2 component3
deb-src protocol://site.example.com/debian distribution component1 component2 component3
```

- deb: paquetes binarios
  - deb-src: paquetes fonte
  - protocol: URL base para a orixe dos paquetes. Pode ser:
    - file:// para indicar a orixe local
    - http:// para indicar a orixe dispoñible nun servidor web
    - ftp:// para indicar a orixe dispoñible nun servidor FTP
    - cdrom:// para indicar a orixe dispoñible nun CD-ROM/DVD-ROM/Blu-ray
  - distribution: code name (stretch, buster, bullseye, sid) ou release class (oldstable, stable, testing, unstable) respectivamente
  - componentX: main ou contrib ou non-free
- DFSG: Directrices de software libre de Debian**
- main: Cumpren DFSG
  - contrib: DFSG pero con dependencias que non cumpren DFSG
  - non-free: Non cumpren DFSG

NOTA: O repositorio stable-backports (buster-backports) contén retroadaptacións de paquetes, é dicir, son paquetes recentes que non existen nunha versión antiga e son recompilados para que poidan ser instalados en estas. Estas retroadaptacións sempre se crean dos paquetes Testing, asegurando así a súa actualización á versión estable cando a seguinte versión estable de Debian atópese dispoñible.

Para poder instalar paquetes de backports debemos indicalo explicitamente con calquera dos 2 comandos seguintes:

```
# apt-get install package/buster-backports
# apt-get install -t buster-backports package
```

Nos comandos anteriores poderíase empregar apt en vez de apt-get.

## Práctica

1. Abrir terminal:
2. Facerse root (su -)
3. Modificar os repositorios, deixando o contido do arquivo */etc/apt/sources.list* como segue:

```
## Debian Buster
deb http://ftp.es.debian.org/debian/ buster main
deb-src http://ftp.es.debian.org/debian/ buster main

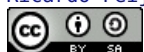
deb http://security.debian.org/ buster/updates main
deb-src http://security.debian.org/ buster/updates main

deb http://ftp.es.debian.org/debian/ buster-updates main
deb-src http://ftp.es.debian.org/debian/ buster-updates main
# Backports repository: testing packages
##deb http://ftp.debian.org/debian buster-backports main contrib non-free
##deb-src http://ftp.debian.org/debian buster-backports main contrib non-free
```

4. Actualizar o sistema. Executar:

```
apt-get update || apt update
apt-get -y upgrade || apt -y upgrade
```

Ricardo Feijoo Costa



This work is licensed under a [Creative Commons Attribution-ShareAlike 4.0 International License](https://creativecommons.org/licenses/by-sa/4.0/)

# Instalación de programas e paquetes

## Práctica: Instalación de paquetes dende repositorios debian mediante apt

```
$ apt #Amosar o uso do comando apt

$ sudo su - #Acceder como root como se fixera login sen sudo. Comúmente empregado nas
ISO LIVE de GNU/Linux para convertirse en root.

# apt update #Actualizar lista de paquetes dispoñibles a instalar dos repositorios

# apt search refcard #Buscar nas descrições dos paquetes o patrón refcard

# apt install debian-refcard #Instalar o paquete debian-refcard

# apt search terminator #Buscar nas descrições dos paquetes o patrón terminator

# apt install terminator #Instalar o paquete terminator

# apt install wine vim wget rsync screen screenfetch telnet netcat tree #Instalar os
paquetes wine, vim, wget, rsync, screen, screenfetch, telnet, netcat, tree

# screenfetch #Executar o comando screenfetch

# tree #Amosar o contido de directorios en forma xerárquica arbórea

# tree -L 1 / #Amosar o contido do directorio / en forma xerárquica arbórea, quedando soamente
no nivel 1 de búsqueda

# tree -L 2 / #Amosar o contido dos 2 primeiros niveis do directorio / en forma xerárquica arbórea

# apt purge debian-refcard #Eliminar e purgar o paquete debian-refcard. É equivalente á
opción remove pero ademais elimina calquera arquivo de configuración do paquete

# echo 'deb http://deb.debian.org/debian buster-backports main contrib non-free' >>
/etc/apt/sources.list #Engadir o repositorio backports á lista de repositorios

# echo 'deb-src http://deb.debian.org/debian buster-backports main contrib non-free'
>> /etc/apt/sources.list #Engadir o repositorio backports de fontes á lista de repositorios

# apt update #Actualizar lista de paquetes dispoñibles a instalar dos repositorios

# apt -y install -t buster-backports checkinstall #Instalar o paquete checkinstall dende
backports

# exit #Pechar a consola de comandos do usuario actual, neste caso do usuario root

$
```

## Práctica: Instalación de paquetes mediante dpkg

```
$ cd /tmp #Acceder ao directorio /tmp

$ apt-get download debian-refcard #Descargar o ficheiro deb do paquete buscado debian-refcard na
ruta actual. Descargará soamente o paquete e non as súas dependencias.

$ sudo su - #Acceder como root como se fixera login sen sudo. Comúmente empregado nas
ISO LIVE de GNU/Linux para convertirse en root.

# cd /tmp #Acceder ao directorio /tmp

# dpkg -i $(ls debian-refcard*) #Instalar o paquete descargado debian-refcard.

# dpkg -l debian-refcard #Listar información sobre o paquete debian-refcard

# dpkg -P debian-refcard #Eliminar e purgar o paquete debian-refcard. É equivalente á opción
remove pero ademais elimina calquera arquivo de configuración do paquete

# exit #Pechar a consola de comandos do usuario actual, neste caso do usuario root

$
```

## Práctica: Instalación mediante compilación de SOURCES (fontes)

```
$ cd /tmp #Acceder ao directorio /tmp

$ wget http://nginx.org/download/nginx-1.17.6.tar.gz #Descargar na ruta actual o ficheiro nginx-1.17.6.tar.gz

$ tar xvfz nginx-1.17.6.tar.gz #Descomprimir o ficheiro nginx-1.17.6.tar.gz

$ cd nginx-1.17.6 #Acceder ao directorio nginx-1.17.6

$ ./configure #Configurar o programa para a súa compilación, é dicir, indicar con que configuración faremos a compilación xerando ao final do proceso un arquivo de nome Makefile. Neste caso a configuración non é posible amosando o erro que a impide.

$ sudo su - #Acceder como root como se fixera login sen sudo. Comúmente empregado nas ISO LIVE de GNU/Linux para convertirse en root.

# apt search pcre #Buscar nas descrições dos paquetes o patrón pcre

# apt install libghc-pcre-light-prof #Instalar o paquete libghc-light-prof

# apt search zlib #Buscar nas descrições dos paquetes o patrón zlib

# apt install zlib1g-dev #Instalar o paquete zlib1g-dev

# exit #Pechar a consola de comandos do usuario actual, neste caso do usuario root

$ ./configure #Configurar o programa para a súa compilación, é dicir, indicar con que configuración faremos a compilación xerando ao final do proceso un arquivo de nome Makefile. Neste caso a configuración é posible xerando un arquivo Makefile.

$ ls -l #Listar de forma extendida o contido do directorio actual

$ make #Compilar nginx a través do Makefile xerado.

$ sudo su - #Acceder como root como se fixera login sen sudo. Comúmente empregado nas ISO LIVE de GNU/Linux para convertirse en root.

# cd /tmp/nginx-1.17.6 #Acceder ao directorio /tmp/nginx-1.17.6

# make install #Instalar nginx.

# /usr/local/nginx/sbin/nginx #Executar nginx

# telnet localhost 80 #Comprobar se temos conectividade co porto TCP 80 (http)

# nc -vz localhost 80 #Comprobar se temos conectividade co porto TCP 80 (http)

# exit #Pechar a consola de comandos do usuario actual, neste caso do usuario root

$
```

## Práctica: Instalación ficheiros .bin ou .run

- Insertar en VirtualBox o ficheiro VirtualGuestAdditions.iso.

```
$ sudo su - #Acceder como root como se fixera login sen sudo. Comúmente empregado nas ISO LIVE de GNU/Linux para convertirse en root.

# mount /dev/sr0 /mnt #Montar o dispositivo /dev/sr0 en /mnt

# cd /mnt #Acceder ao directorio /mnt

# ls -l #Listar de forma extendida o contido do directorio actual

# file VBoxLinuxAdditions.run
VBoxLinuxAdditions.run: POSIX shell script executable (binary data) #Amosar o tipo de ficheiro que se indica como parámetro

# ./VBoxLinuxAdditions.run #Executar o script de instalación VBoxLinuxAdditions.run.

# exit #Pechar a consola de comandos do usuario actual, neste caso do usuario root

$
```

## Práctica: Instalación ficheiros .exe mediante wine

```
$ cd /tmp #Acceder ao directorio /tmp

$ wget https://download.geogebra.org/package/win-autoupdate -O Geogebra-Windows-
Installer-6-0-564-0.exe #Descargar o ficheiro .exe de Geogebra e gardalo co nome Geogebra-Windows-
Installer-6-0-564-0.exe

$ ls -l #Listar de forma extendida o contido do directorio actual

$ wine Geogebra-Windows-Installer-6-0-564-0.exe #Instalar mediante wine o ficheiro .exe de
Geogebra. O instalador crea (a primeira vez que se executa wine) ou garda configuracións e programas
dentro da ruta ~/.wine propia ao usuario que executa o comando. O instalador atopa dependencias e
permite instalalas.
```

## Práctica: Instalación ficheiros portables (supostamente soamente habería que executalos)

```
$ wget https://download.geogebra.org/package/linux-port6 -O Geogebra-Linux64-Portable-6-
0-564-0.zip #Descargar para arquitecturas 64bit o ficheiro portable .zip de Geogebra e gardalo co nome
Geogebra-Linux64-Portable-6-0-564-0.zip

$ wget https://download.geogebra.org/package/linux-port6-32 -O Geogebra-Linux32-
Portable-6-0-564-0.zip #Descargar para arquitecturas 32bit o ficheiro portable .zip de Geogebra e
gardalo co nome Geogebra-Linux32-Portable-6-0-564-0.zip

$ unzip Geogebra-Linux32-Portable-6-0-564-0.zip #Descomprimir o ficheiro arquitectura 32bit
Geogebra-Linux32-Portable-6-0-564-0.zip

$ cd GeoGebra-linux-ia32 #Acceder ao directorio GeoGebra-linux-ia32

$ ls -l #Listar de forma extendida o contido do directorio actual

$ ls -l GeoGebra #Listar de forma extendida o ficheiro GeoGebra

$ ./GeoGebra #Executar Geogebra. Este comando amosa erro e non permite a execución do programa
porque faltan dependencias por cumprir, é dicir, no portable descargado non están autocontidas as
librarías que se necesitan, polo que o programa non pode ser executado. Isto soe ser normal nos métodos
de instalación que non revisan as dependencias necesarias.

$ sudo su - #Acceder como root como se fixera login sen sudo. Comúmente empregado nas
ISO LIVE de GNU/Linux para convertirse en root.

    # apt search libgconf #Buscar nas descrições dos paquetes o patrón libgconf

    # apt install libgconf-2-4 #Instalar o paquete libgconf-2-4

    # exit #Pechar a consola de comandos do usuario actual, neste caso do usuario root

$ ./GeoGebra #Executar GeoGebra. Este comando amosa erro aínda que agora permite a execución do
programa.
```

Ricardo Feijoo Costa



This work is licensed under a **Creative Commons Attribution-ShareAlike 4.0 International License**



# systemd, systemctl, /etc/init.d

## URLs de interese

- [Tarxeta de Referencia para Debian GNU/Linux](#)
- [Servizos UNIX](#)

## Práctica

```
# apt-get update #Actualizar repositorios

# apt-cache search openssh | grep server #Buscar paquetes openssh

# apt-cache search ^ssh server #Buscar paquetes cos patróns ssh e server

# apt-get install openssh-server #Instalar paquete openssh-server

# dpkg -l openssh-server #Listar información sobre o paquete openssh-server

# systemctl #Listar todos os servizos en execución

# systemctl | grep ssh #Buscar referencias sobre o servizo ssh

# systemctl status ssh.service #Ver o estado do servizo ssh

# /etc/init.d/ssh status #Equivale ao comando anterior

# service ssh status #Equivale ao comando anterior

# systemctl stop ssh.service #Parar o servizo ssh

# /etc/init.d/ssh stop #Equivale ao comando anterior

# service ssh stop #Equivale ao comando anterior

# systemctl start ssh.service #Arrancar o servizo ssh

# /etc/init.d/ssh start #Equivale ao comando anterior

# service ssh start #Equivale ao comando anterior

# systemctl enable ssh.service #Permite que o servizo ssh sexa iniciado no arranque xerando os links nos runlevels (/etc/rcX.d)

# find /etc/rc* -name "*ssh*" #Busca polas links runlevels nos cartafolios /etc/rc*

# systemctl disable ssh.service #Deshabilita o servizo ssh para que non sexa iniciado no arranque xerando/modificando os links nos runlevels (/etc/rcX.d)

# find /etc/rc* -name "*ssh*" #Busca polas links runlevels nos cartafolios /etc/rc*

# systemctl is-enabled ssh.service #Amosa se o servizo ssh está enabled ou disabled
```

## Práctica

```
# cd /lib/systemd/system && ls #Amosa os ficheiros de systemd na ruta que son manipulados polo sistema

# cd /etc/systemd/system && ls #Amosa os ficheiros de systemd na ruta que poden ser manipulados polo administrador

# cd /etc/systemd/system && ls -l sshd.service #Amosa a ligazón simbólica do service ssh que apunta a
/lib/systemd/system/ssh.service

# cd /etc/systemd/system && cat sshd.service #Amosa a contido do service ssh

# cd /lib/systemd/system && cat ssh.service #Equivale ao comando anterior xa que o arquivo sshd.service é unha ligazón ao
arquivo ssh.service

# cd /lib/systemd/system && ls *.service #Amosa os services de systemd

# cd /lib/systemd/system && ls *.target #Amosa os targets de systemd

# cd /lib/systemd/system && ls -l runlevel?.target #Amosa os targets equivalentes aos runlevels

# find /lib/systemd/system -name "runlevel?.target" | xargs -I VAR bash -c 'echo VAR ; echo --- ; cat VAR ; echo --- ; \
sleep 10' #Atopa os targets equivalentes aos runlevels e amosa o seu contido esperando 10 segundos entre cada arquivo

# find /lib/systemd/system -name "runlevel?.target" | sort | xargs -I VAR bash -c 'echo VAR ; echo --- ; grep -i description \
VAR ; echo ---' #Atopa os targets equivalentes aos runlevels na orde ascendente de 0 a 6 e amosa o valor do patrón buscado
Description

# cd /lib/systemd/system && diff3 runlevel[2-4].target #Busca diferencias entre os targets equivalentes aos runlevels 2,3 e 4. Non
atopa diferencias xa que os 3 arquivos son ligazón ao mesmo target: multi-user.target

# cd /lib/systemd/system && ls -l runlevel[2-4].target #Os 3 arquivos listados son ligazóns simbólicas ao mesmo target: multi-
user.target

# cd /etc/systemd/system && ls -ld *target.wants #Amosa os cartafoles dos targets que posúen dependencias para o seu correcto
funcionamento

# cd /etc/systemd/system && ls -l *target.wants #Amosa as ligazóns dos cartafoles dos targets que posúen dependencias para o
seu correcto funcionamento, é dicir, amosa que é o que precisan os targets(requires) para o seu correcto funcionamento

# cd /etc/systemd/system && ls -lah $(ls -lah runlevel?.target | awk '{print $NF}' | sed 's/$/.wants/g' | sort -u | xargs) #Amosa
as dependencias de cada runlevel para o seu correcto funcionamento

# grep -Hi requires $(find /lib/systemd/system -type d -name "*.wants" | xargs -I VAR ls VAR) | sort -u #Amosa dependencias
a través dos Requires de cada ficheiro de configuración
```

Ricardo Feijoo Costa



This work is licensed under a [Creative Commons Attribution-ShareAlike 4.0 International License](https://creativecommons.org/licenses/by-sa/4.0/)

# cron, atd

**cron:** /etc/crontab (global) , /var/spool/cron/crontabs/username

/etc/crontab (global)

minuto hora día mes día-semana usuario script(comando/s)



Imaxe tomada de <http://www.linuxconfig.org>

## Variables de entorno (man 5 crontab)

Pódense definir variables como **SHELL, PATH, MAILTO**:

```
SHELL=/bin/bash
PATH=/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin
MAILTO=pepito, pepito@example.local, user@gmail.com
##onde:
#####pepito é un usuario local
```

Pódense empregar para cada un dos campos: o carácter \* (todos os valores dende o primeiro ao último pertimido), listas(carácter ,), rangos(carácter -), saltos(carácter /) e combinación destes. Así, para o campo minutos:

- **8,11**: Execútase a tarefa neses minutos, é dicir, no minuto 8 e no 11.
- **8-11**: Execútase a tarefa en tódolos minutos comprendidos entre o 8 e o 11, estes incluído, é dicir, nos minutos 8, 9, 10 e 11.
- **0-4,8-12**: Combinación dos 2 anteriores. Executarase nos minutos: 0, 1, 2, 3, 4, 8, 9, 10, 11, 12
- **\*/15**: Execútase a tarefa executarase cada 15 minutos.
- **0,15,30,45**: Equivale á liña anterior.
- **0-59/10**: Execútase cada 10 minutos dende o minuto 0 ao 59, é dicir, nos minutos 0, 10, 20, 30, 40 e 50.
- **0,10,20,30,40,50**: Equivale á liña anterior.

## Práctica

17 \* \* \* \* root cd /tmp && example1.sh #Execútase como o usuario root o comando "cd /tmp && example1.sh"  
tódolos días a todas horas pasados 17 minutos de cada hora, é dicir, ás 00:17, 01:17, 02:17 ...23:17

17 4 \* \* \* root cd /tmp && example1.sh #Execútase como o usuario root o comando "cd /tmp && example1.sh"

tódolos días ás 04:17h

```
17 4 12 * * root cd /tmp && example1.sh #Execútase como o usuario root o comando "cd /tmp && example1.sh"
```

tódolos días 12 de cada mes ás 04:17h

```
17 4 12 2 * root cd /tmp && example1.sh #Execútase como o usuario root o comando "cd /tmp && example1.sh"
```

tódolos 12 de Febreiro ás 04:17h

```
17 4 12 2 7 root cd /tmp && example1.sh #Execútase como o usuario root o comando "cd /tmp && example1.sh"
```

tódolos 12 de Febreiro sempre e cando sexa Domingo ás 04:17h

```
17 4 12 2 0 root cd /tmp && example1.sh #Equivale ao comando anterior
```

```
* * * * * root /tmp/example2.sh #Execútase como o usuario root o comando "/tmp/example2.sh" en cada minuto
```

```
* * * * * root /bin/bash /tmp/example2.sh #Execútase como o usuario root o comando "/bin/bash /tmp/example2.sh"
```

en cada minuto

```
0 0 1 1 * root /bin/bash /tmp/example2.sh #Execútase como o usuario root o comando "/bin/bash /tmp/example2.sh" o 1 de Xaneiro ás 00:00. Equivale a @yearly
```

@yearly root /bin/bash /tmp/example2.sh #Equivale ao comando anterior.

@annually root /bin/bash /tmp/example2.sh #Equivale ao comando anterior.

```
0 0 1 * * root /bin/bash /tmp/example2.sh #Execútase como o usuario root o comando "/bin/bash /tmp/example2.sh" o día 1 de cada mes ás 00:00. Equivale a @monthly
```

@monthly root /bin/bash /tmp/example2.sh #Equivale ao comando anterior.

```
0 0 * * 0 root /bin/bash /tmp/example2.sh #Execútase como o usuario root o comando "/bin/bash /tmp/example2.sh" tódolos domingos ás 00:00. Equivale a @weekly
```

@weekly root /bin/bash /tmp/example2.sh #Equivale ao comando anterior.

```
0 0 * * * root /bin/bash /tmp/example2.sh #Execútase como o usuario root o comando "/bin/bash /tmp/example2.sh" tódolos días ás 00:00. Equivale a @daily
```

@daily root /bin/bash /tmp/example2.sh #Equivale ao comando anterior.

@midnight root /bin/bash /tmp/example2.sh #Equivale ao comando anterior.

```
0 * * * * root /bin/bash /tmp/example2.sh #Execútase como o usuario root o comando "/bin/bash /tmp/example2.sh" cada hora. Equivale a @hourly
```

@hourly root /bin/bash /tmp/example2.sh #Equivale ao comando anterior.

## `/var/spool/cron/crontabs/username`

`minuto hora día mes día-semana script(comando/s)`

Todos os exemplos anteriores de `/etc/crontab` son válidos, soamente hai que ter en conta que no `crontab` de cada usuario existe unha columna menos. Así, os comandos anteriores son válidos eliminando a columna 6(respectiva ao usuario) do `/etc/crontab`.

```
$ crontab -l #Lista as tarefas programadas do usuario
```

```
$ crontab -e #Abre editor por defecto para poder modificar/crear as tarefas programadas do usuario. O editor por defecto é gardado na variable EDITOR, polo cal pódese modificar/crear esta variable para definir o editor por defecto para cron.
```

```
$ export EDITOR='/usr/bin/vim'; crontab -e #Abre o editor vim para poder modificar/crear as tarefas programadas do usuario.
```

## `atd`

O comando **at** permite executar un script(comando/s) nun tempo determinado, obtendo a data e hora desexadas como parámetros.

```
# at 20:00 -f /tmp/emplo3.sh #Executa o script "/tmp/emplo3.sh" ás 20:00h
```

```
# at 20:00 12.02.16 -f /tmp/emplo3.sh #Executa o script "/tmp/emplo3.sh" o día 12 de Febreiro de 2016 ás 20:00h
```

```
# at now + 1 minutes -f /tmp/emplo3.sh #Executa o script "/tmp/emplo3.sh" no próximo minuto a partir de agora, é dicir, se o comando foi lanzado ás 19:56:32 executará a tarefa programada ás 19:57:00.
```

```
# at now + 15 minutes
```

```
at> reboot <Ctrl>+<d>
```

```
at> <EOT> #Reinicia o sistema pasados 15 minutos do instante actual
```

```
# at now + 2 hours
```

```
at> reboot <Ctrl>+<d>
```

```
at> <EOT> #Reinicia o sistema pasadas 2 horas do instante actual
```

```
# at 4pm + 3 days <Enter>
```

```
at> tar cvjf backup.tar.bz2 /var/www/site-example1
```

```
at> <Ctrl>+<d>
```

```
at> <EOT> #Pasados 3 días do actual e ás 16:00h executar o comando "tar cvjf backup.tar.bz2 /var/www/site-example1"
```

```
# at 15:00 <<EOF
```

```
> echo "Tarefa realizada" | mail pepito@example.local
```

```
> EOF #Envía un correo á dirección pepito@example.local as 15:00h
```

```
# atq #Lista a cola de tarefas programadas por at
```

# atrm numberid #Elimina a tarefa programada co identificador numberid

Ricardo Feijoo Costa



This work is licensed under a [Creative Commons Attribution-ShareAlike 4.0 International License](https://creativecommons.org/licenses/by-sa/4.0/)

# Kernel: Parámetros de arranque





**ESCENARIO:** Para verificar o que acontece nas prácticas crear en VirtualBox 2 máquinas virtuais coas seguintes características:

1. MÁQUINA VIRTUAL 1:
  - Nome: **DebianKernel64**
  - RAM: 2048MB
  - Procesador: 2 ao 100%
  - ISO: Live Debian 64bits, escritorio XFCE
  - 1 disco duro dinámico:
    - Nome: **DebianKernel64.vdi**
    - Tamaño: 20GB
    - Ten instalado o SO GNU/Linux Debian 10 de 64bits.
      - Nome de usuario: usuario
      - Nome computador: usuario-pc
      - Contraseñal: abc123. (Olla que o contraseñal ten un carácter punto final)
  - Primeira opción de arranque: Óptica
  - Segunda opción de arranque: Disco duro
2. MÁQUINA VIRTUAL 2:
  - Nome: **Live64**
  - RAM: 2048MB
  - Procesador: 2 ao 100% e PAE activado.
  - ISO: Live Debian 64bits, escritorio XFCE
  - Primeira opción de arranque: Óptica

## Práctica1

1. Arrancar a máquina virtual 1 creada en modo Inicio normal

**VBoxManage startvm DebianKernel64**

2. Ao ter como primeira opción de arranque Óptica detense o arranque do sistema operativo no xestor de arranque **SYSLINUX (isolinux)**, na súa primeira opción, á espera de escoller unha opción.
3. En calquera opción premer a tecla Tabulado  para ver os parámetros de arranque co cal arranxa o kernel.
4. Non modificar nada, escoller a primeira opción e premer a tecla Intro  para arrancar con eses parámetros do kernel.
5. Na contorna gráfica (shell xfce) abrir un terminal e executar:

\$ setxkbmap es #Cambiar o mapa de teclado ao idioma español.

\$ cat /proc/cmdline #Amosar o contido de /proc/cmdline que contén os parámetros cos cales o kernel foi arrancado.

\$ init 6 #Comando para enviar o runlevel (nivel de execución) do sistema operativo ao nivel 6, equivalente a reiniciar o sistema. Trala execución deste comando amosarase un erro, debido a que o usuario sen permisos de root (administrador) non accede ao comando init, é dicir, non atopa o comando init no PATH que ten definido para a execución dos comandos..



\$ echo \$PATH #Ver o PATH definido para este usuario.

\$ whereis init #O comando whereis localiza os ficheiros binarios, fontes e páxinas do manual correspondentes a un programa, neste caso buscamos a información para o comando init. Atopamos que o comando está localizado en /usr/sbin/, a cal é unha ruta que non está definida no PATH do usuario (ver comando anterior echo \$PATH).

\$ /usr/sbin/init 6 #Comando para enviar o runlevel (nivel de execución) do sistema operativo ao nivel 6, equivalente a reiniciar o sistema. Agora como usuario sen privilexios de administrador podemos enviar a máquina ao nivel 6 de arranque (reiniciar) xa que é posible executar ese comando nesa ruta sen os privilexios de root.

6. Voltar a cargar o Live CD, é dicir, voltar a cargar o escenario da práctica.



7. Ao ter como primeira opción de arranque Óptica detense o arranque do sistema operativo no xestor de arranque **SYSLINUX (isolinux)**, na súa primeira opción, á espera de escoller unha opción.
8. Na primeira opción premer a tecla Tabulado  para ver os parámetros de arranque co cal arranxa o kernel.
9. Eliminar as opcións **quiet** e **splash** e premer a tecla Intro  para arrancar sen eses parámetros do kernel. Agora no arranque veremos todas as mensaxes que amosa o kernel para chegar a arrancar o sistema operativo.

10. Na contorna gráfica (shell xfce) abrir un terminal e executar:


\$ setxkbmap es #Cambiar o mapa de teclado ao idioma español.

\$ cat /proc/cmdline #Amosar o contido de /proc/cmdline que contén os parámetros cos cales o kernel foi arrancado.

\$ /usr/sbin/init 6 #Comando para enviar o runlevel (nivel de execución) do sistema operativo ao nivel 6, equivalente a reiniciar o sistema. Agora como usuario sen privilexios de administrador podemos enviar a máquina ao nivel 6 de arranque (reiniciar) xa que é posible executar ese comando nesa ruta sen os privilexios de root.

11. Voltar a cargar o Live CD, é dicir, voltar a cargar o escenario da práctica.



12. Ao ter como primeira opción de arranque Óptica detense o arranque do sistema operativo no xestor de arranque **SYSLINUX (isolinux)**, na súa primeira opción, á espera de escoller unha opción.
13. Na primeira opción premer a tecla Tabulado  para ver os parámetros de arranque co cal arranxa o kernel.

14. Eliminar as opcións **quiet** e **splash**.



15. Premer unha vez a **barra espaciadora** para conseguir un espazo `␣`, engadir a continuación **init=/bin/bash** e premer a tecla Intro `↵` para arrancar con novos parámetros do kernel. Agora no arranque veremos que non chegamos a arrancar o sistema operativo porque o primeiro proceso a chamar (init ou systemd) está modificado a /bin/bash, co cal en vez de facer unha chamada ao arranque do sistema operativo facemos unha chamada a unha consola de comandos, polo que, accedemos a unha consola onde temos permisos de root (administrador). **Olló!: Non está cargado completamente o sistema operativo, pero si está recoñecido o hardware.**

16. Executar:

```
# cat /proc/cmdline #Amosar o contido de /proc/cmdline que contén os parámetros cos cales o kernel foi arrancado.
```

```
# reboot -f #Reiniciar de forma forzosa, é dicir, reiniciar se pechar o sistema de xeito seguro.
```

## Práctica2

Realizar de novo a Práctica1 pero agora no sistema operativo instalado no disco duro (ver escenario). Agora, teremos un novo xestor de arranque: **GRUB versión 2 ou GRUB 2**



1. Pór como primeira opción de arranque o Disco duro.
2. O xestor de arranque GRUB 2 arranca por defecto na súa primeira opción en 5segundos. Entón, parar o arranque deste primeira opción premendo as teclas frechas abaixo ↓, arriba ↑.
3. Seleccionar a primeira opción de arranque.
4. Premer a tecla **e** (edit) para poder editar os parámetros de arranque do kernel.
5. Moverse coa tecla frecha abaixo ↓ ata chegar á liña onde aparecen os parámetros **ro quiet splash**
6. Substituír os parámetros **ro quiet splash** polos parámetros **rw init=/bin/bash**. e premer as teclas **<Ctrl> + x**, é dicir, `^x`, para arrancar a opción escollida con novos parámetros do kernel. Agora no arranque veremos que non chegamos a arrancar o sistema operativo porque o primeiro proceso a chamar (init ou systemd) está modificado a /bin/bash, co cal en vez de facer unha chamada ao arranque do sistema operativo facemos unha chamada a unha consola de comandos, polo que, accedemos a unha consola onde temos permisos de root (administrador). **Olló!: Non está cargado completamente o sistema operativo, pero si está recoñecido o hardware.**

7. Executar:

```
# cat /proc/cmdline #Amosar o contido de /proc/cmdline que contén os parámetros cos cales o kernel foi arrancado.
```

```
# passwd usuario #Modificar o contrasinal do usuario de nome usuario. Pór como contrasinal 1234. Repetir o contrasinal. Olló: Non aparecen asteriscos nin outro tipo de caracteres para impedir saber cantos e cales caracteres estamos a escribir.
```

```
# reboot -f #Reiniciar de forma forzosa, é dicir, reiniciar se pechar o sistema de xeito seguro.
```

8. Deixar arrancar a opción por defecto e comprobar que agora o usuario de nome *usuario* ten o contrasinal modificado a 1234

Ricardo Feijoo Costa



This work is licensed under a **Creative Commons Attribution-ShareAlike 4.0 International License**



# SHELL BASH: /bin/bash

## Startup files

### man 1 bash (INVOCATION)

Cando se inicia bash lense unha serie de ficheiros de arranque, os cales, cargan variables de entorno, o prompt do sistema (PS1), o PATH, alias...

- bash como login shell interactivo: cando entramos no sistema con login e password, usamos **su -**, ou iniciamos bash coa opción **--login**

/etc/profile (lese sempre) -->

--> ~/.bash\_profile --> ~/.bash\_login --> ~/.profile (busca no orde exposta e lese o primeiro que exista) -->

--> ~/.bash\_logout (lese ao saír)

- bash como non-login shell interactivo: cando se abre unha nova consola sen login e password, usamos **su**, ou cando se inicia bash sen opcións

/etc/bash.bashrc (se existe) ---> ~/.bashrc (se existe)

- bash como shell non interactivo: cando se executa nun script busca pola variable **BASH\_ENV** e expande o seu valor, e usa este valor como un ficheiro a ler e executar.

```
if [ -n "$BASH_ENV" ]; then . "$BASH_ENV"; fi
```

### No caso de debian: shell interactivo

/etc/profile --> /etc/bash.bashrc --> ~/.profile --> ~/.bashrc --> ~/.bash\_logout

Ricardo Feijoo Costa



This work is licensed under a [Creative Commons Attribution-ShareAlike 4.0 International License](https://creativecommons.org/licenses/by-sa/4.0/)