

# 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         0 -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         0 -20     0     0     0 S   0,0  0,0   0:00.00 kworker/0:0+
   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/)

# Permisos ugo: u=usuario, g=grupo, o=outros

## chmod, chown, chgrp, umask

- **chmod**: Cambia os permisos de acceso de ficheiros. Nunca cambia os permisos das ligazóns simbólicas pero si dos ficheiros aos que apuntan.
- **chown**: Cambia o usuario e grupo propietario de ficheiros
- **chgrp**: Cambia o grupo ao que pertencen os ficheiros
- **umask**: Estable a máscara de permisos e directorios por defecto á hora de crealos. Non cambia permisos de ficheiros e directorios existentes.

## Máscara de permisos

- Cando listamos en forma extendida (ls -l) ficheiros e directorios, amósanse os permisos **ugo** correspondentes, é dicir, amósase a continuación do tipo de ficheiro unha máscara de 9 permisos, que actúan de 3 en 3. Así, na primeira terna os permisos pertencen ao **usuario propietario**, na segunda terna os permisos pertencen ao **grupo propietario**, e na terceira terna os permisos pertencen aos **outros** (resto do mundo, aqueles usuarios e grupos que non son o usuario propietario nin o grupo propietario). Por exemplo:

```
$ ls -l /etc/passwd /usr/bin/passwd
-rw-r--r-- 1 root root 2122 oct 16 21:52 /etc/passwd
-rwsr-xr-x 1 root root 63736 jul 27 2018 /usr/bin/passwd
```

Permiso	Caracter	Número base octal
lectura	r	4
escritura	w	2
execución	x	1
sen permiso	-	0
execución+SETUID	<b>s</b> (sustitúe ao x na terna do usuario propietario)	4000
sen execución+SETUID	<b>S</b> (sustitúe ao x na terna do usuario propietario)	4000
execución+SETGID	<b>s</b> (sustitúe ao x na terna do grupo propietario)	2000
sen execución+SETGID	<b>S</b> (sustitúe ao x na terna do grupo propietario)	2000
execución+Sticky bit	<b>t</b> (sustitúe ao x na terna de outros)	1000
sen execución+Sticky bit	<b>T</b> (sustitúe ao x na terna de outros)	1000

## Máscara de permisos por defecto (umask)

- Cando creamos un ficheiro ou directorio por defecto seranlle asignados un permisos. Estes permisos veñen predefinidos mediante o comando **umask**. Están predefinidos nos arquivos /etc/login.defs, /etc/profile e ~/.profile

## chmod

Pódese empregar de 2 formas: mediante caracteres ou mediante números en octal.

### Sintaxe do comando mediante caracteres:

**chmod argumento file**

**argumento = [ugoa][+|=][rwxXstugo]** : Posible argumento a empregar para modificar permisos.

```
$ chmod [ugoa] + [rwxXstugo] file #Engadir permiso/s.  
$ chmod [ugoa] - [rwxXstugo] file #Quitar permiso/s.  
$ chmod [ugoa] = [rwxXstugo] file #Soamente este/s permiso/s.
```

#### [ugoa]

**u** equivale a **usuario propietario**. Para modificar na terna de permisos pertencente ao usuario propietario.

**g** equivale a **grupo propietario**. Para modificar na terna de permisos pertencente ao grupo propietario.

**o** equivale a **outros (resto do mundo)**. Para modificar na terna de permisos pertencente aos outros (resto do mundo).

**a** equivale a **todos (u+g+o)**. Para modificar nas 3 ternas de permisos.

#### [rwxXstugo]

**r** equivale a **lectura**

**w** equivale a **escritura**

**x** equivale a **execución para un ficheiro e permiso de acceso para directorios**

**X** equivale a **execución soamente se o ficheiro é un directorio ou xa ten permiso de execución para algún usuario**

**s** equivale a **execución e activar SETUID(4000) ou SETGID(2000)**

**t** equivale a **execución e activar Sticky bit(1000)**

**-** equivale a **Non posúe permiso**

### Sintaxe do comando mediante números en octal:

**chmod argumento file**

**[argumento] = [n<sub>4</sub>][n<sub>3</sub>][n<sub>2</sub>][n<sub>1</sub>]** : Posible argumento a empregar para modificar permisos. Como mínimo débese empregar un número e como máximo 4 números.

**n<sub>4</sub>** : Número correspondente aos permisos SUID(4), SGID(2) ou Sticky bit(1), ou a ausencia de permiso(0), ou calquera suma dos anteriores. Por exemplo:

5 = 4 + 0 + 1, co cal os permisos serían de SUID e Sticky: **s-t**

7 = 4 + 2 + 1, co cal os permisos serían de SUID, SGID e Sticky bit: **sst**

**n<sub>3</sub>** : Número correspondente aos permisos pertencentes á terna do usuario propietario.

**n<sub>2</sub>** : Número correspondente aos permisos pertencentes á terna do grupo propietario.

**n<sub>1</sub>** : Número correspondente aos permisos pertencentes á terna dos outros (resto do mundo).

Nas 3 ternas os números poden ser:

- **4(r)**, **2(w)**, **1(x)** ou **0(-)**(Sen permiso).

- Calquera combinación de suma das anteriores. Por exemplo:

5 = 4 + 0 + 1, co cal os permisos serían de lectura e execución: **r-x**

7 = 4 + 2 + 1, co cal os permisos serían de lectura, escritura e execución: **rwX**

```
$ chmod n1 file #Cambiar permiso/s á terna pertencente aos outros (resto do mundo).
```

```
$ chmod n2n1 file #Cambiar permiso/s ás ternas pertencentes ao grupo propietario e aos outros (resto do mundo).
```

```
$ chmod n3n2n1 file #Cambiar permiso/s ás ternas pertencentes ao usuario propietario, ao grupo propietario e aos outros (resto do mundo).
```

```
$ chmod n4n3n2n1 file #Cambiar permiso/s SUID e/ou SGID e/ou Sticky bit e ás ternas pertencentes ao usuario propietario, ao grupo propietario e aos outros (resto do mundo).
```

## Práctica chmod: Sintaxe do comando mediante caracteres

\$ rm -rf /tmp/permisos /tmp/permisos/subdir1 #Eliminar /tmp/permisos e /tmp/permisos/subdir1

\$ mkdir -p /tmp/permisos/subdir1 #Crear cartafol /tmp/permisos e /tmp/permisos/subdir1

\$ for i in \$(seq 1 5); do touch /tmp/permisos/\$i.txt; done #Crear 5 ficheiros baleiros en /tmp/permisos: 1.txt, 2.txt, 3.txt, 4.txt e 5.txt

\$ for i in \$(seq 1 2); do touch /tmp/permisos/subdir1/in\${i}.txt; done #Crear 2 ficheiros baleiros en /tmp/permisos/subdir1: in1.txt e in2.txt

\$ ls -lR /tmp/permisos #Listar de forma extendida o contido do cartafol /tmp/permisos e de forma recursiva tódolos subdirectorios atopados.

\$ ls -ld /tmp/permisos #Listar soamente os permisos do cartafol /tmp/permisos, é dicir, listar os permisos do propio cartafol pero non os do seu contido.

\$ ls -ld /tmp/permisos/subdir1 #Listar soamente os permisos do cartafol /tmp/permisos/subdir1, é dicir, listar os permisos do propio cartafol pero non os do seu contido.

A continuación imos executar comandos **chmod**. Para comprobar os permisos que se modifican, a continuación de cada execución dun comando **chmod** executaremos os 3 comandos anteriores: `ls -lR /tmp/permisos`, `ls -ld /tmp/permisos` e `ls -ld /tmp/permisos/subdir1`

\$ chmod a-r /tmp/permisos/\*.txt #Eliminar permisos de lectura a todas as ternas, é dicir, eliminar permisos de lectura para ugo (usuario propietario, grupo propietario e outros) nos ficheiros 1.txt, 2.txt, 3.txt, 4.txt e 5.txt

Repetición dos 3 comandos ls anteriores

\$ chmod u+r /tmp/permisos/\*.txt #Engadir permisos de lectura ao usuario propietario dos ficheiros 1.txt, 2.txt, 3.txt, 4.txt e 5.txt

Repetición dos 3 comandos ls anteriores

\$ chmod -R a+rwX /tmp/permisos #Engadir todos os permisos(rwx) ao directorio /tmp/permisos e a tódolos ficheiros e subdirectorios pertencentes a /tmp/permisos, é dicir, otorgar permisos de lectura, escritura e execución a /tmp/permisos, a 1.txt, 2.txt, 3.txt, 4.txt e 5.txt pertencentes a /tmp/permisos, a /tmp/permisos/subdir1 e aos ficheiros in1.txt e in2.txt pertencentes a /tmp/permisos/subdir1. Debido á opción -R (recursividade) o comando efectúase no directorio /tmp/permisos e en tódolos subdirectorios atopados.

Repetición dos 3 comandos ls anteriores

\$ chmod -R g-w /tmp/permisos #Eliminar todos os permisos de escritura(w) ao directorio /tmp/permisos e a tódolos ficheiros e subdirectorios pertencentes a /tmp/permisos, é dicir, eliminar permisos de escritura ao directorio /tmp/permisos, e a 1.txt, 2.txt, 3.txt, 4.txt e 5.txt pertencentes a /tmp/permisos, ao directorio /tmp/permisos/subdir1 e aos ficheiros in1.txt e in2.txt pertencentes a /tmp/permisos/subdir1. Debido á opción -R (recursividade) o comando efectúase no directorio /tmp/permisos e en tódolos subdirectorios atopados.

Repetición dos 3 comandos ls anteriores

\$ chmod uo-w /tmp/permisos/\*.txt #Eliminar permisos de escritura(w) a 1.txt, 2.txt, 3.txt, 4.txt e 5.txt pertencentes a /tmp/permisos para o usuario propietario e para outros (resto do mundo).

Repetición dos 3 comandos ls anteriores

\$ chmod uo+w,g-x /tmp/permisos/\*.txt #Otorgar dentro de /tmp/permisos/ nos ficheiros 1.txt, 2.txt, 3.txt, 4.txt e 5.txt o permiso de escritura(w) para o usuario propietario e o permiso de escritura(w) para outros (resto do mundo), e eliminar o permiso de execución(x) para o grupo propietario.

Repetición dos 3 comandos ls anteriores

\$ chmod uo-w,g+x /tmp/permisos #Eliminar ao directorio /tmp/permisos e non ao seu contido o permiso de escritura(w) para o usuario propietario e o permiso de escritura(w) para outros (resto do mundo). Tamén permite otorgar o permiso de execución(x) para o grupo propietario.

Repetición dos 3 comandos ls anteriores

\$ chmod ugo+x /tmp/permisos/\*.txt #Equivale ao comando: `chmod a+x /tmp/permisos/*.txt`, é dicir, agregar dentro de /tmp/permisos/ a 1.txt, 2.txt, 3.txt, 4.txt e 5.txt o permiso de execución(x) para o usuario propietario, o grupo propietario e para outros (resto do mundo).

Repetición dos 3 comandos ls anteriores

\$ chmod u+s,g+s,o+t /tmp/permisos/\*.txt #Agregar dentro de /tmp/permisos/ a 1.txt, 2.txt, 3.txt, 4.txt e 5.txt o permiso de SUID(4000) para o usuario propietario, o permiso de SGID(2000) para o grupo propietario e o permiso Sticky bit(1000) para outros (resto do mundo).

Repetición dos 3 comandos ls anteriores

\$ **chmod ugo-x /tmp/permisos/\*.txt** #Equivale ao comando: **chmod a-x /tmp/permisos/\*.txt**, é dicir, eliminar dentro de /tmp/permisos/ a 1.txt, 2.txt, 3.txt, 4.txt e 5.txt o permiso de execución(x) para o usuario propietario, o grupo propietario e para outros (resto do mundo).

**Repetición dos 3 comandos ls anteriores**

\$ **chmod ug-sx-r+w,o+t-x,g+sx /tmp/permisos/\*.txt** #Modificar os permisos do directorio /tmp/permisos/ e non do seu contido. Así:

- Para o usuario propietario(u): u-sx-r+w, co cal elimínase o permiso SUID(s), o permiso de execución(x) e o permiso de lectura, e agrégase o permiso de escritura.
- Para o grupo propietario(g): g-sx-r+w+sx, co cal +sx anula a -sx. Así, agrégase o permiso de escritura(w), SUID(s) e execución(x), e elimínase o permiso de lectura(r).
- Para os outros(o): o+t, co cal agrégase o permiso Sticky bit(t).

**Repetición dos 3 comandos ls anteriores**

## Práctica chmod: Sintaxe do comando mediante números en octal

\$ **chmod -R 777 /tmp/permisos** #Engadir todos os permisos(rwx) ao directorio /tmp/permisos e a tódolos ficheiros e subdirectorios pertencentes a /tmp/permisos, é dicir, otorgar permisos de lectura, escritura e execución a /tmp/permisos, a 1.txt, 2.txt, 3.txt, 4.txt e 5.txt pertencentes a /tmp/permisos, a /tmp/permisos/subdir1 e aos ficheiros in1.txt e in2.txt pertencentes a /tmp/permisos/subdir1. Debido á opción -R (recursividade) o comando efectúase no directorio /tmp/permisos e en tódolos subdirectorios atopados.

**Repetición dos 3 comandos ls anteriores**

\$ **chmod -R 4750 /tmp/permisos** #Engadir o permiso SUID(4000) e todos os permisos(rwx) ao usuario propietario do directorio /tmp/permisos e a tódolos ficheiros e subdirectorios pertencentes a /tmp/permisos, os permisos de lectura e execución(5=4+0+1=r-x) ao grupo propietario de /tmp/permisos e de tódolos ficheiros e subdirectorios pertencentes a /tmp/permisos, e ningún permiso(0=---) a terna de permisos outros (resto do mundo) de /tmp/permisos e de tódolos ficheiros e subdirectorios pertencentes a /tmp/permisos.

**Repetición dos 3 comandos ls anteriores**

\$ **chmod -R 7750 /tmp/permisos** #Engadir o permiso SUID(4000), o permiso SGID(2000) e o permiso Sticky bit(1000) e todos os permisos(rwx) ao usuario propietario do directorio /tmp/permisos e a tódolos ficheiros e subdirectorios pertencentes a /tmp/permisos, os permisos de lectura e execución(5=4+0+1=r-x) ao grupo propietario de /tmp/permisos e de tódolos ficheiros e subdirectorios pertencentes a /tmp/permisos, e ningún permiso(0=---) a terna de permisos outros (resto do mundo) de /tmp/permisos e de tódolos ficheiros e subdirectorios pertencentes a /tmp/permisos.

**Repetición dos 3 comandos ls anteriores**

## Práctica: chmod e ligazóns simbólicas

\$ **ln -s /tmp/permisos/1.txt /tmp/permisos/lsimb1.txt** #Crear a ligazón simbólica /tmp/permisos/lsimb1.txt, é dicir, crear un acceso directo /tmp/permisos/lsimb1.txt que apunte a /tmp/permisos/1.txt

\$ **ls -li /tmp/permisos/1.txt /tmp/permisos/lsimb1.txt** #Listar de forma extendida cos inodos o ficheiro /tmp/permisos/1.txt e a ligazón simbólica /tmp/permisos/lsimb1.txt

\$ **chmod u-r /tmp/permisos/lsimb1.txt** #Non cambia os permisos da ligazón simbólica lsimb1.txt senón os permisos do ficheiro a que apunta 1.txt, é dicir, elimina o permiso de lectura ao usuario propietario do ficheiro 1.txt

\$ **ls -li /tmp/permisos/1.txt /tmp/permisos/lsimb1.txt** #Listar de forma extendida cos inodos o ficheiro /tmp/permisos/1.txt e a ligazón simbólica /tmp/permisos/lsimb1.txt

## Práctica: chmod e ligazóns duras

\$ **ln /tmp/permisos/1.txt /tmp/permisos/ldura1.txt** #Crear a ligazón dura /tmp/permisos/ldura1.txt, é dicir, crear un novo nome ao ficheiro 1.txt no mesmo inodo.

\$ **ls -li /tmp/permisos/1.txt /tmp/permisos/ldura1.txt** #Listar de forma extendida cos inodos o ficheiro /tmp/permisos/1.txt e a ligazón dura /tmp/permisos/ldura1.txt

\$ **chmod u+r /tmp/permisos/ldura1.txt** #Otorga o permiso de lectura ao usuario propietario da ligazón dura ldura1.txt

\$ **ls -li /tmp/permisos/1.txt /tmp/permisos/ldura1.txt** #Listar de forma extendida cos inodos o ficheiro /tmp/permisos/1.txt e a ligazón dura /tmp/permisos/ldura1.txt



## Práctica chown

\$ ls -l /tmp/permisos/1.txt #Listar de forma extendida o ficheiro /tmp/permisos/1.txt

\$ ls -lR /tmp/permisos #Listar de forma extendida o contido do cartafol /tmp/permisos e de forma recursiva tódolos subdirectorios atopados.

\$ ls -ld /tmp/permisos #Listar soamente os permisos do cartafol /tmp/permisos, é dicir, listar os permisos do propio cartafol pero non os do seu contido.

A continuación imos executar comandos **chown**. Para comprobar os permisos que se modifican, logo de cada execución dun comando **chown** executaremos os comandos ls anteriores: ls -l /tmp/permisos/1.txt, ls -lR /tmp/permisos e ls -ld /tmp/permisos

\$ sudo su - #Acceder á consola de root(administrador) a través dos permisos configurados co comando sudo (/etc/sudoers, visudo)

# chown root /tmp/permisos/1.txt #Cambiar o usuario propietario de /tmp/permisos/1.txt ao usuario root

Repetición dos comandos ls anteriores

# chown root.root /tmp/permisos/1.txt #Cambiar o usuario propietario e o grupo propietario de /tmp/permisos/1.txt ao usuario root e ao grupo root

Repetición dos comandos ls anteriores

# chown root. /tmp/permisos/1.txt #Equivale ao comando anterior

Repetición dos comandos ls anteriores

# chown root: /tmp/permisos/1.txt #Equivale ao comando anterior

Repetición dos comandos ls anteriores

# chown .usuario /tmp/permisos/1.txt #Cambiar o grupo propietario de /tmp/permisos/1.txt ao grupo usuario

Repetición dos comandos ls anteriores

# chown :usuario /tmp/permisos/1.txt #Equivale ao comando anterior

Repetición dos comandos ls anteriores

# chown -R root. /tmp/permisos #Cambiar o usuario propietario a root e o grupo propietario a root ao directorio /tmp/permisos e recursivamente a tódolos ficheiros e directorios contidos en /tmp/permisos.

Repetición dos comandos ls anteriores

# chown -R usuario. /tmp/permisos #Cambiar o usuario propietario a usuario e o grupo propietario a usuario ao directorio /tmp/permisos e recursivamente a tódolos ficheiros e directorios contidos en /tmp/permisos.

Repetición dos comandos ls anteriores

# exit #Saír da consola local de root para voltar á consola local do usuario usuario.

\$

## Práctica chgrp

\$ ls -l /tmp/permisos/1.txt #Listar de forma extendida o ficheiro /tmp/permisos/1.txt

\$ ls -lR /tmp/permisos #Listar de forma extendida o contido do cartafol /tmp/permisos e de forma recursiva tódolos subdirectorios atopados.

\$ ls -ld /tmp/permisos #Listar soamente os permisos do cartafol /tmp/permisos, é dicir, listar os permisos do propio cartafol pero non os do seu contido.

A continuación imos executar comandos **chgrp**. Para comprobar os permisos que se modifican, logo de cada execución dun comando **chgrp** executaremos os comandos ls anteriores: ls -l /tmp/permisos/1.txt, ls -lR /tmp/permisos e ls -ld /tmp/permisos

\$ sudo su - #Acceder á consola de root(administrador) a través dos permisos configurados co comando sudo (/etc/sudoers, visudo)

# chgrp root /tmp/permisos/1.txt #Cambiar o grupo propietario de /tmp/permisos/1.txt ao grupo root

Repetición dos comandos ls anteriores

# chgrp -R root /tmp/permisos #Cambiar o grupo propietario a root ao directorio /tmp/permisos e recursivamente a tódolos ficheiros e directorios contidos en /tmp/permisos.

Repetición dos comandos ls anteriores

# exit #Saír da consola local de root para voltar á consola local do usuario usuario.

\$

## Práctica umask

\$ umask #Amosa a máscara de permisos para a creación de ficheiros e directorios en formato número octal (ver chmod). O primeiro dígito é sempre '0' e pódese empregar ou non.

\$ umask -S #Amosa a máscara de permisos para a creación de ficheiros e directorios en formato carácter (ver chmod).

\$ umask 0055 #Cambia a máscara de permisos para a creación de ficheiros e directorios en formato octal, determinando que agora os ficheiros creados terán os permisos 622(rw- -w- -w-), e os directorios terán os permisos 722(rwx -w- -w-).

Para obter os permisos dos ficheiros e directorios debemos facer a seguinte operación co comando umask:

1. Os ficheiros considérase que posúen por defecto permisos 666 (rw- rw- rw-)  
Os directorios considérase que posúen por defecto permisos 777 (rwx rwx rwx)
2. Pásase a binario os permisos do comando umask, é dicir, cámbiase os valores da máscara a binario.
3. Faise o Complemento a 1 deses valores, é dicir, os 0 pasan ser 1 e viceversa
4. Faise a operación AND cos valores 666 en binario(110 110 110) pertencentes aos ficheiros ou 777(111 111 111) pertencentes aos directorios

Exemplo:

### ■ Ficheiros:

	Permisos	Binario	Operación
Ficheiros	666	110 110 110	110 110 110
umask	055	000 101 101	Ca1: 111 010 010
AND: 110 010 010			

Así, os permisos actuais son: 622

### ■ Directorios:

	Permisos	Binario	Operación
Directorios	777	111 111 111	111 111 111
umask	055	000 101 101	Ca1: 111 010 010
AND: 111 010 010			

Así, os permisos actuais son: 722

\$ umask -S #Amosa a máscara de permisos para a creación de ficheiros e directorios en formato carácter (ver chmod).

\$ mkdir -p /tmp/dir2 #Crear o cartafol /tmp/dir2

\$ touch /tmp/f1.txt /tmp/dir2/f2.txt #Crear os ficheiros /tmp/f1.txt e /tmp/dir2/f2.txt

\$ ls -ld /tmp/dir2 #Listar soamente os permisos do cartafol /tmp/dir2, é dicir, listar os permisos do propio cartafol pero non os do seu contido.

\$ ls -l /tmp/f1.txt /tmp/dir2/f2.txt #Listar de forma extendida os ficheiros /tmp/f1.txt e /tmp/dir2/f2.txt.

Ricardo Feijoo Costa



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



# Outros permisos distintos a ugo: Atributos.

**lsattr, chattr: Atributos 'aAcCdDeiJsStTu'**

- **append only (a):** Soamente pódese abrir o ficheiro para engadir escritura. Este atributo soamente o pode engadir/quitar o usuario root.
- **no atime updates (A):** Non se actualiza atime: Marca temporal sobre a última lectura dun ficheiro. Por exemplo, cando se accede a ler con cat, more...
- **compressed (c):** Antes de escribir un ficheiro este comprímese no disco polo kernel. Antes de ler descomprímese.
- **no copy on write (C):** Activar/desactivar copy-on-write(COW) en ficheiros e directorios. Co atributo dado estará desactivado o copy-on-write.
- **no dump (d):** Cando se execute dump non se fará backup do ficheiro
- **synchronous directory updates (D):** Os cambios no cartafol escríbense de forma sincrónica(inmediatamente) no disco.
- **extent format (e):** É un atributo do sistema de ficheiros (ext4 o posúe). Activo indica que o ficheiro está empregando extents. Un extent é basicamente un grupo de bloques físicamente contiguos, o cal mellora o rendemento.
- **immutable (i):** O ficheiro é immutable e polo tanto non pode ser modificado. Así non pode ser editado nin eliminado, nin se poden crear ligazóns duras, nin pode ser sobreescrito. Este atributo soamente o pode engadir/quitar o usuario root.
- **data journalling (j):** En sistemas de ficheiros ext3 e ext4 se son montados coas opcións data=ordered ou data=writeback será escrito no journal, e se son montados coa opción data=journal non posúe efecto. Este atributo soamente o pode engadir/quitar o usuario root.
- **secure deletion (s):** Con este atributo o ficheiro será eliminado enchendo o contido dos seus bloques a cero.
- **synchronous updates (S):** Equivale á opción sync de mount, entón os cambios son escritos inmediatamente en disco. É equivalente, para os ficheiros, ao atributo D dos cartafoles.
- **no tail-merging (t):** Non válido en sistemas de ficheiros ext2, ext3. Con este atributo os ficheiros non presentan fragmentación.
- **top of directory hierarchy (T):** Activa nos cartafoles o Orlov block allocator. Así, escribirase nas zonas máis rápidas do disco.
- **and undeletable (u):** Cando o ficheiro é eliminado o seu contido é gardado podendo entón ser recuperado.

## BUGS E LIMITACIÓNS:

- Os atributos 'c', 's' e 'u' non son soportados polos sistemas de ficheiros ext2, ext3 e ext4
- O atributo 'j' soamente é empregado nos sistemas de ficheiros ext3 e ext4
- O atributo 'D' soamente é empregado en kernel Linux 2.5.19 ou superiores

**\$ chattr + [atributos] file : Engadir atributos.**

**\$ chattr - [atributos] file : Quitar atributos.**

**\$ chattr = [atributos] file : Soamente estes atributos.**

**# chattr + [atributos] file : Engadir atributos.**

**# chattr - [atributos] file : Quitar atributos.**

**# chattr = [atributos] file : Soamente estes atributos.**

## Práctica

```
$ rm -rf /tmp/atributos #Eliminar /tmp/atributos

$ mkdir /tmp/atributos #Crear cartafol /tmp/atributos

$ for i in $(seq 1 5); do touch /tmp/atributos/$i.txt; done #Crear 5 ficheiros baleiros en /tmp/atributos: 1.txt, 2.txt, 3.txt, 4.txt e 5.txt

$ ls -l /tmp/atributos #Listar de forma extendida o contido do cartafol /tmp/atributos

$ lsattr /tmp/atributos #Listar os atributos do contido do cartafol /tmp/atributos

$ lsattr -d /tmp/atributos #Listar soamente os atributos do cartafol /tmp/atributos, é dicir, listar os atributos do propio cartafol pero non os do seu contido.

$ mkdir -p /tmp/atributos/sub1 /tmp/atributos/sub2 #Crear cartafoles /tmp/atributos/sub1 e /tmp/atributos/sub2

$ for i in $(seq 1 4); do touch /tmp/atributos/sub1/sub1_${i}.txt; done #Crear 4 ficheiros baleiros dentro de /tmp/atributos/sub1: sub1_1.txt, sub1_2.txt, sub1_3.txt e sub1_4.txt

$ lsattr /tmp/atributos #Listar os atributos do contido do cartafol /tmp/atributos quedándose no
```

primeiro nivel, é dicir, listar os atributos dos ficheiros e cartafol que se verían co comando `ls /tmp/atributos`

`$ lsattr -d /tmp/atributos` #Listar soamente os atributos do cartafol /tmp/atributos, é dicir, listar os atributos do propio cartafol pero non os do seu contido.

`$ lsattr -R /tmp/atributos` #Listar os atributos de toda a estrutura dentro de /tmp/atributos, é dicir, listar os atributos de tódolos ficheiros e cartafol contidos en /tmp/atributos, pero non lista os atributos do propio cartafol /tmp/atributos

`$ lsattr -Ra /tmp/atributos` #Listar os atributos de toda a estrutura de /tmp/atributos, incluído todos aqueles que comecen co carácter punto, é dicir, listar os atributos de tódolos ficheiros e cartafol contidos en /tmp/atributos incluídos os ocultos e ao propio cartafol /tmp/atributos (.) como ao pai /tmp (..)

`$ echo $(date) > /tmp/atributos/1.txt` #Sobreescribe o contido do ficheiro /tmp/atributos/1.txt, sendo o contido a saída do comando `date`

`$ cat /tmp/atributos/1.txt` #Amosa o contido do ficheiro /tmp/atributos/1.txt, sendo o contido a saída do comando anterior: `date`

`$ chattr +i /tmp/atributos/1.txt` #Erro. Non permitido para usuarios. Soamente root pode engadir/quitar o permiso "i" (immutable)

`$ su - -c "chattr +i /tmp/atributos/1.txt"` #Executando mediante o usuario root: Incorpora o atributo "i" (immutable) ao arquivo /tmp/atributos/1.txt. Agora ese arquivo non pode editarse ou eliminarse ata que deixe de ter o atributo immutable.

`$ lsattr /tmp/atributos` #Listar os atributos do contido do cartafol /tmp/atributos

`$ echo $(date) > /tmp/atributos/1.txt` #Erro: Permiso denegado. Intenta sobreescribir o contido do ficheiro /tmp/atributos/1.txt, sendo o contido a saída do comando `date`, pero debido a que posúe o atributo immutable non pode modificarse.

`$ echo $(date) >> /tmp/atributos/1.txt` #Erro: Permiso denegado. Intenta engadir no contido do ficheiro /tmp/atributos/1.txt, a saída do comando `date`, pero debido a que posúe o atributo immutable non pode modificarse.

`$ rm -f /tmp/atributos/1.txt` #Erro: Operación non permitida. Intenta eliminar o contido do ficheiro /tmp/atributos/1.txt, pero debido a que posúe o atributo immutable non pode modificarse.

`$ ln /tmp/atributos/1.txt /tmp/atributos/ldura1.txt` #Erro: Operación non permitida. Intenta crear unha ligazón dura (mesmo inodo, nome distinto) do ficheiro /tmp/atributos/1.txt, pero debido a que posúe o atributo immutable non pode crearse.

`$ ln -s /tmp/atributos/1.txt /tmp/atributos/lsimb.txt` #Créase unha ligazón simbólica (distinto ficheiro, distinto inodo) do ficheiro /tmp/atributos/1.txt.

`$ su - -c "chattr -i /tmp/atributos/1.txt"` #Executando mediante o usuario root: Elimina o atributo "i" (immutable) ao arquivo /tmp/atributos/1.txt, polo cal agora o arquivo pode modificarse.

`$ lsattr /tmp/atributos` #Listar os atributos do contido do cartafol /tmp/atributos

`$ echo $(date) > /tmp/atributos/1.txt` #Sobreescribe o contido do ficheiro /tmp/atributos/1.txt, sendo o contido a saída do comando `date`. Debido a que agora non posúe o atributo immutable o arquivo pode modificarse.

`$ echo $(date) >> /tmp/atributos/1.txt` #Engade a saída do comando `date` ao contido do ficheiro /tmp/atributos/1.txt. Debido a que agora non posúe o atributo immutable o arquivo pode modificarse.

`$ cat /tmp/atributos/1.txt` #Amosa o contido do ficheiro /tmp/atributos/1.txt

`$ rm -f /tmp/atributos/1.txt` #Elimina o ficheiro /tmp/atributos/1.txt

`$ ls -l /tmp/atributos/1.txt` #Erro: O arquivo non existe.

`$ lsattr -d /tmp/atributos/sub2` #Listar soamente os atributos do cartafol /tmp/atributos/sub2, é dicir, listar os atributos do propio cartafol pero non os do seu contido.

`$ touch /tmp/atributos/sub2/sub2_A.txt` #Crear o ficheiro baleiro /tmp/atributos/sub2/sub2\_A.txt

`$ echo $(date) > /tmp/atributos/sub2/sub2_A.txt` #Sobreescribe o contido do ficheiro /tmp/atributos/sub2/sub2\_A.txt, sendo o contido a saída do comando `date`

`$ su - -c "chattr +i /tmp/atributos/sub2"` #Executando mediante o usuario root: Incorpora o atributo "i" (immutable) ao cartafol /tmp/atributos/sub2

\$ lsattr -d /tmp/atributos/sub2 #Listar soamente os atributos do cartafol /tmp/atributos/sub2, é dicir, listar os atributos do propio cartafol pero non os do seu contido.

\$ touch /tmp/atributos/sub2/sub2\_B.txt #Erro: Permiso denegado. Intenta escribir dentro do cartafol /tmp/atributos/sub2, pero debido a que posúe o atributo immutable non se pode.

\$ echo \$(date) > /tmp/atributos/sub2/sub2\_A.txt #Sobreescribe o contido do ficheiro /tmp/atributos/sub2/sub2\_A.txt, sendo o contido a saída do comando date, xa que aínda que o cartafol está immutable o ficheiro xa existía e non era immutable.

\$ echo \$(date) >> /tmp/atributos/sub2/sub2\_A.txt #Engade contido ao ficheiro /tmp/atributos/sub2/sub2\_A.txt, xa que aínda que o cartafol está immutable o ficheiro xa existía e non era immutable.

\$ cat /tmp/atributos/sub2/sub2\_A.txt #Amosa o contido do ficheiro /tmp/atributos/sub2/sub2\_A.txt

\$ rm -f /tmp/atributos/sub2/sub2\_A.txt #Erro: Permiso denegado. Intenta eliminar o ficheiro /tmp/atributos/sub2/sub2\_A.txt, pero non pode debido a que aínda que o ficheiro non está immutable o cartafol si o está.

\$ rm -rf /tmp/atributos/sub2 #Erro: Permiso denegado. Intenta eliminar o cartafol /tmp/atributos/sub2, pero non pode xa que o cartafol posúe o atributo immutable.

\$ su - -c "chattr -i /tmp/atributos/sub2" #Executando mediante o usuario root: Quita o atributo "i" (immutable) ao cartafol /tmp/atributos/sub2.

\$ lsattr -d /tmp/atributos/sub2 #Listar soamente os atributos do cartafol /tmp/atributos/sub2, é dicir, listar os atributos do propio cartafol pero non os do seu contido.

\$ touch /tmp/atributos/sub2/sub2\_B.txt #Crear o ficheiro baleiro /tmp/atributos/sub2/sub2\_B.txt

\$ echo \$(date) > /tmp/atributos/sub2/sub2\_B.txt #Sobreescribe o contido do ficheiro /tmp/atributos/sub2/sub2\_B.txt, sendo o contido a saída do comando date

\$ cat /tmp/atributos/sub2/sub2\_B.txt #Amosa o contido do ficheiro /tmp/atributos/sub2/sub2\_B.txt, sendo o contido a saída do comando anterior: date

\$ chattr +a /tmp/atributos/sub2/sub2\_B.txt #Erro. Non permitido para usuarios. Soamente root pode engadir/quitar o permiso "a" (append only)

\$ su - -c "chattr +a /tmp/atributos/sub2/sub2\_B.txt" #Executando mediante o usuario root: Incorpora o atributo "a" (append only) ao arquivo /tmp/atributos/sub2/sub2\_B.txt.

\$ lsattr -Ra /tmp/atributos #Listar os atributos de toda a estrutura de /tmp/atributos, incluído todos aqueles que comecen co carácter punto, é dicir, listar os atributos de tódolos ficheiros e cartafoles contidos en /tmp/atributos incluídos os ocultos e ao propio cartafol /tmp/atributos (.) como ao pai /tmp (..)

\$ lsattr /tmp/atributos/sub2/sub2\_B.txt #Listar os atributos de /tmp/atributos/sub2/sub2\_B.txt

\$ echo \$(date) > /tmp/atributos/sub2/sub2\_B.txt #Erro: Permiso denegado. Intenta sobreescribir o ficheiro /tmp/atributos/sub2/sub2\_B.txt, pero debido a que posúe o atributo append only (a) non se pode.

\$ echo \$(date) >> /tmp/atributos/sub2/sub2\_B.txt #Engade a saída do comando date ao contido do ficheiro /tmp/atributos/sub2/sub2\_B.txt. Debido a que posúe o atributo append only (a) no arquivo, este soamente pódese abrir para engadir escritura.

\$ cat /tmp/atributos/sub2/sub2\_B.txt #Amosa o contido do ficheiro /tmp/atributos/sub2/sub2\_B.txt

\$ ls -l /tmp/atributos/sub2/sub2\_B.txt #Listar de forma extendida o ficheiro /tmp/atributos/sub2/sub2\_B.txt

\$ sleep 60 && touch /tmp/atributos/sub2/sub2\_B.txt #Espera 60 segundos e logo modifica a data do ficheiro /tmp/atributos/sub2/sub2\_B.txt, pero non o seu contido.

\$ ls -l /tmp/atributos/sub2/sub2\_B.txt #Listar de forma extendida o ficheiro /tmp/atributos/sub2/sub2\_B.txt

\$ rm -f /tmp/atributos/sub2/sub2\_B.txt #Erro: Operación non permitida. Intenta eliminar o contido do ficheiro /tmp/atributos/sub2/sub2\_B.txt, pero debido a que posúe o atributo append only (a) non pode modificarse, soamente está aberto para engadir escritura.

\$ su - -c "chattr -a /tmp/atributos/sub2/sub2\_B.txt" #Executando mediante o usuario root: Quita o atributo "a" (append only) ao arquivo /tmp/atributos/sub2/sub2\_B.txt.

```
$ lsattr /tmp/atributos/sub2/sub2_B.txt #Listar os atributos de /tmp/atributos/sub2/sub2_B.txt
```

```
$ rm -f /tmp/atributos/sub2/sub2_B.txt #Elimina o arquivo /tmp/atributos/sub2/sub2_B.txt
```

```
$ ls -l /tmp/atributos/sub2/sub2_B.txt #Erro: O arquivo non existe.
```

---

[Ricardo Feijoo Costa](#)



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

# Outros permisos distintos a ugo: as ACL (Access Control List)

## Preferencia de permisos (de maior a menor):

- **Atributos→uo de ugo→ACLs**  
En caso de conflito entre o **usuario propietario/otros (uo de ugo)** e as **ACLs** **prevalecen os permisos uo de ugo**
- **Atributos→máscara ACLs→g de ugo e outros usuarios e grupos distintos de ug (de ugo)**  
En caso de conflito entre o **grupo propietario(g de ugo)/otros grupos distintos do propietario/otros usuarios distintos do propietario** e as **ACLs** **prevalecen as ACLs**

## Soporte para ACLs nos sistemas de ficheiros

Hoxe en día o kernel trae incorporado por defecto soporte para ACLs para distintos sistemas de ficheiros. Podemos verificalo co seguinte comando:

```
$ [ -f /boot/config-$(uname -r) ] && grep -i acl /boot/config-$(uname -r)
CONFIG_EXT4_FS_POSIX_ACL=y
CONFIG_REISERFS_FS_POSIX_ACL=y
CONFIG_JFS_POSIX_ACL=y
CONFIG_XFS_POSIX_ACL=y
CONFIG_BTRFS_FS_POSIX_ACL=y
CONFIG_FS_POSIX_ACL=y
CONFIG_TMPFS_POSIX_ACL=y
...
```

Pero no caso que así non sexa debemos activar no sistema de ficheiros o soporte para as ACLs, polo que deberiamos instalar o paquete **acl** e modificar o arquivo **/etc/fstab**:

```
# apt-get update && apt-get -y install acl
# cat /etc/fstab | nl
...
7 # <file system> <mount point> <type> <options> <dump> <pass>
8 # / was on /dev/sda1 during installation
9 UUID=3e1ae11e-dac7-4a58-8aec-d06a345171dc / ext4 acl,errors=remount-ro 0 1
...
```

- No **cuarto campo** do ficheiro **/etc/fstab** correspondente aos opcións de montaxe debemos agregar a opción **acl** e posteriormente debemos remontar o sistema de ficheiros modificado. Para non ter que reiniciar podemos empregar calquera dos 2 seguintes comandos:  

```
# mount -a #Remonta todos os sistemas de ficheiros seguindo a orde en /etc/fstab
# mount -o remount /dev/sda1 #Remonta soamente o sistema de ficheiros modificado en /etc/fstab (en este caso /dev/sda1)
```

## Comandos para ACLs: getfacl, setfacl

- **getfacl**: Comando que permite obter o estado das ACLs dos ficheiros e directorios. Opcións de interese:
  - R → Recursividade. Amosa de forma recursiva a todos os ficheiros e directorios pertencentes a un directorio
  - d → Herdable. Amosa soamente as entradas herdables das ACLs.
  - e → Efectivos. Amosa os permisos efectivos en comentarios, aínda se son idénticos aos permisos definidos pola entrada ACL.

Se empregamos o comando **ls -l** apreciaremos en todos aqueles ficheiros e directorios aos que se lle apliquen ACLs o signo **+** Se o sistema de ficheiros onde o comando é empregado non soporta ACLs amosaranse os permisos ugo. O formato da saída do comando segue esta estrutura:

```
$ ls -ld 4acls/
drwsr-xr-x+ 2 alumno primaria 4096 feb 20 19:25 4acls/
$ getfacl 4acls/
1 # file: 4acls/
2 # owner: alumno
3 # group: primaria
4 # flags: s--
5 user::rwx
6 user:pepito:rwx #effective:r-x
7 group::rwx #effective:r-x
8 group:secundaria:r-x
9 mask::r-x
10 other::r-x
11 default:user::rwx
12 default:user:pepito:rwx #effective:r-x
13 default:group::r-x
14 default:mask::r-x
15 default:other:---
```

As liñas **1--3** indican: nome ficheiro, usuario propietario e grupo propietario.  
A liña **4** indica os bits setuid (s), setgid (s), e sticky (t): se existe a letra entón o bit está activo, senón aparece un guión. Soamente aparece esta liña cando algún dos bits está activo.  
As liñas **5, 7 e 10** corresponden aos permisos ugo  
As liñas **6 e 8** son as entradas ACL de usuario e grupo.  
A liña **9** é a máscara. Esta entrada limita os permisos efectivos que se aplican para todos os grupos e todos os usuarios distintos do usuario propietario (u de ugo). **O usuario propietario e os outros (uo de ugo) non son afectados pola máscara, pero o resto das entradas si (Ver Preferencia de permisos (de maior a menor)).**  
As liñas **11--15** amosan as ACLs por defecto asociadas con este directorio. Os directorios poden ter ACL por defecto, é dicir, cando se crea un ficheiro/directorio dentro de este directorio aplicánselle as ACLs por defecto (son as ACLs herdables). Os ficheiros nunca teñen unha ACL por defecto.

O comportamento para getfacl é amosar as ACL e as ACL por defecto(herdables), e engadir os permisos efectivos comentados en liñas onde os permisos das entradas difiren dos permisos efectivos. Os listados de ACL de múltiples ficheiros sepáranse por liñas en branco. A saída de getfacl pode ser usada como entrada para setfacl

- **setfacl:** Comando que permite modificar o estado das ACLs dos ficheiros e directorios. Opcións de interese:

- m → Permite crear unha ACL
- x → Permite eliminar unha ACL
- R → Permite que a ACL sexa recursiva, é dicir, afecta de forma recursiva a todos os ficheiros e directorios pertencentes a un directorio
- d → Permite que a ACL sexa herdable, é dicir, afecta aos novos ficheiros e directorios que se creen nun futuro pertencentes a un directorio
- b → Elimina todas as ACLs xeradas dun directorio, pero non elimina os permisos ugo, nin elimina as ACLs do contido dese directorio
- Rb → Elimina todas as ACLs xeradas
- u:username:rwX → Indica que a ACL afectará a un usuario de nome username cos permisos rwX. En permisos se algún non existe pode omitirse ou ben substituírse con 1 guión. Así nunha ACL, r-X é o mesmo que rX
- u::rwX → Indica que a ACL afectará ao usuario propietario (ugo) cos permisos rwX. En permisos se algún non existe pode omitirse ou ben substituírse con 1 guión. Así nunha ACL, r-X é o mesmo que rX
- g:groupname:rwX → Indica que a ACL afectará a un grupo de nome groupname cos permisos rwX. En permisos se algún non existe pode omitirse ou ben substituírse con 1 guión. Así nunha ACL, r-X é o mesmo que rX
- g::rwX → Indica que a ACL afectará ao grupo propietario (ugo) cos permisos rwX. En permisos se algún non existe pode omitirse ou ben substituírse con 1 guión. Así nunha ACL, r-X é o mesmo que rX
- o::rwX → Indica que a ACL afectará á terna de permisos outros (ugo) cos permisos rwX. En permisos se algún non existe pode omitirse ou ben substituírse con 1 guión. Así nunha ACL, r-X é o mesmo que rX
- m:r-X → Permite modificar a máscara de permisos por defecto (Ver getfacl). En permisos se algún non existe pode omitirse ou ben substituírse con 1 guión. Así nunha ACL, r-X é o mesmo que rX



## Crear Escenario

- Xerar 2 grupos: primaria e secundaria; xerar 2 usuarios: ana e brais, sendo ana pertencente ao grupo primaria e brais ao grupo secundaria

```
# for i in primaria secundaria;do groupadd $i;done
# useradd -m -d /home/ana -s /bin/bash -g primaria -p $(mkpasswd 123456) ana
# useradd -m -d /home/brais -s /bin/bash -g secundaria -p $(mkpasswd 654321) brais
```

## Práctica1: Conflicto permisos ugo-ACL (Ver Preferencia de permisos (de maior a menor))

En caso de conflito entre o **usuario propietario/otros (u de ugo)** e as **ACLs** **prevalecen os permisos uo de ugo**

1. Xerar o cartafol /revisar pertencente ao usuario ana e grupo primaria cos permisos 550:  

```
# rm -rf /revisar && mkdir /revisar && chown ana.primaria /revisar && chmod 550 /revisar
```
2. Cos permisos ugo o usuario ana non pode crear ficheiros/cartafoles dentro de /revisar, co cal imos crear a seguinte ACL:  

```
# setfacl -m u:ana:rwX /revisar #Segue ana sen ter permiso de xerar nada xa que uo prevalece sobre a ACL sen afectarlle deste xeito a máscara efectiva de permisos
# getfacl /revisar
getfacl: Eliminando '/' inicial en nombres de ruta absolutos
# file: revisar/
# owner: ana
# group: primaria
user::r-x
user:ana:rwX #effective:rwX
group::r-x #effective:r-x
mask::rwX
other::---
# su - ana
$ cd /revisar
$ touch file1.txt
touch: no se puede efectuar `touch' sobre «file1.txt»: Permiso denegado
```

## Práctica2: Conflicto permisos ugo-ACL (Ver Preferencia de permisos (de maior a menor))

En caso de conflito entre o **grupo propietario(g de ugo)/otros grupos distintos do propietario/otros usuarios distintos do propietario** e as **ACLs** **prevalecen as ACLs**

1. Xerar o cartafol /revisar pertencente ao usuario ana e grupo primaria cos permisos 550:  

```
# rm -rf /revisar && mkdir /revisar && chown ana.primaria /revisar && chmod 550 /revisar
```
2. Cos permisos ugo o usuario brais que pertence ao grupo secundaria non pode crear ficheiros/cartafoles dentro de /revisar, co cal imos crear a seguinte ACL:  

```
# setfacl -m g:secundaria:rwX /revisar #Agora brais ten permiso de xerar contido dentro de /revisar xa que a ACLs prevalece sobre os permisos de calquera grupo e calquer usuario distinto do propietario de /revisar, é dicir, a máscara efectiva de permisos prevalece sobre calquera grupo ou sobre calquera usuario que non sexa o propietario de /revisar (u de ugo)
# getfacl /revisar
getfacl: Eliminando '/' inicial en nombres de ruta absolutos
# file: revisar
# owner: ana
# group: primaria
user::r-x
group::r-x #effective:r-x
group:secundaria:rwX #effective:rwX
mask::rwX
other::---
# su - brais
$ cd /revisar
$ touch file1.txt
```

### Práctica3: ACLs

1. Xerar o cartafol /comunPRI pertencente ao grupo root e ao usuario root cos permisos ugo 700:  
# mkdir /comunPRI && chown root. /comunPRI && chmod 700 /comunPRI
2. Xerar o cartafol /comunPRI/ana pertencente ao usuario ana e ao grupo primaria cos permisos ugo 700:  
# mkdir /comunPRI/ana && chown ana.primaria /comunPRI/ana && chmod 700 /comunPRI/ana
3. Xerar o cartafol /comunPRI/resto pertencente ao grupo primaria e ao usuario root cos permisos ugo 750:  
# mkdir /comunPRI/resto && chown root.primaria /comunPRI/resto && chmod 750 /comunPRI/resto
4. Crear as seguintes ACLs para os cartafoles /comunPRI, /comunPRI/ana e /comunPRI/resto (Vaise comprobar de cada vez as ACLs xeradas co comando getfacl -R /comunPRI) e intentar acceder cos usuarios ana e brais de cada vez:

#### Escenario

```
ana ∈ grupo primaria
brais ∈ grupo secundaria

/comunPRI
ana ten permisos ugo --- en /comunPRI
brais ten permisos ugo --- en /comunPRI
grupo primaria ten permisos ugo --- en /comunPRI
grupo secundaria ten permisos ugo --- en /comunPRI

/comunPRI/ana
ana ten permisos ugo rwx en /comunPRI/ana
brais ten permisos ugo --- en /comunPRI/ana
grupo primaria ten permisos ugo --- en /comunPRI/ana
grupo secundaria ten permisos ugo --- en /comunPRI/ana

/comunPRI/resto
ana ten permisos ugo r-x en /comunPRI/resto por pertencer ao grupo primaria
brais ten permisos ugo --- en /comunPRI/resto
grupo primaria ten permisos ugo r-x en /comunPRI/resto
grupo secundaria ten permisos ugo --- en /comunPRI/resto

# getfacl -R /comunPRI/
getfacl: Eliminando '/' inicial en nombres de ruta absolutos
# file: comunPRI/
# owner: root
# group: root
user::rwx
group::---
other::---

# file: comunPRI/ana
# owner: ana
# group: primaria
user::rwx
group::---
other::---

# file: comunPRI/resto
# owner: root
# group: primaria
user::rwx
group::r-x
other::---
```

- A. Que o grupo primaria teña permiso de lectura en /comunPRI e /comunPRI/ana, e permiso de escritura herdable en /comunPRI/resto

```
# su - ana
$ cd /comunPRI
$ -su: cd: /comunPRI/: Permiso denegado
# setfacl -m g:primaria:rx /comunPRI
# setfacl -m g:primaria:rx /comunPRI/ana
# setfacl -dm g:primaria:rwx /comunPRI/resto
# getfacl -R /comunPRI
# su - ana
$ cd /comunPRI #Agora ana ten acceso (probar con brais)
```

- B. Que ana poida acceder e ver o contido en /comunPRI e que poida modificar o contido dos cartfoles /comunPRI/resto e /comunPRI/ana

```
# setfacl -m u:ana:rx /comunPRI #Esta ACL non é necesaria porque ana xa tiña permisos ugo rx sobre este cartafol por pertencer ao grupo primaria
# setfacl -m u:ana:rwx /comunPRI/ana #Esta ACL non é necesaria porque ana xa tiña permisos ugo rwx sobre este cartafol
# setfacl -m u:ana:rwx /comunPRI/resto #Esta ACL é necesaria porque ana soamente tiña permisos ugo rx sobre este cartafol por pertencer ao grupo primaria
# getfacl -R /comunPRI
```

- C. Que o grupo secundaria só poida acceder e ver o contido en /comunPRI, pero que non poida modificar nada nin acceder aos cartafoles /comunPRI/ana e /comunPRI/resto

```
# setfacl -m g:secundaria:rx /comunPRI
# setfacl -m g:secundaria:--- /comunPRI/ana
# setfacl -m g:secundaria:--- /comunPRI/resto
# getfacl -R /comunPRI
# su - brais
$ cd /comunPRI #Agora brais ten acceso por pertencer ao grupo secundaria
$ cd ana #brais non ten acceso debido á ACL creada (pertencer ao grupo secundaria)
-su: cd: ana/: Permiso denegado
$ cd resto #brais non ten acceso debido á ACL creada (pertencer ao grupo secundaria)
-su: cd: resto/: Permiso denegado
```

D. Que brais aínda que pertenza ao grupo secundaria non poida acceder a /comunPRI

```
# setfacl -m u:brais:--- /comunPRI #Agora brais non terá acceso
# getfacl -R /comunPRI
# su - brais
$ -su: cd: /comunPRI/: Permiso denegado
```

E. Eliminar a ACL referente a brais

```
# setfacl -x u:brais /comunPRI
# getfacl -R /comunPRI
```

F. Que brais poida acceder a todos os cartafolés de /comunPRI e ver o seu contido pero non modificalos e ademais que eses permisos sexan herdables para futuros ficheiros ou directorios que se creen dentro de /comunPRI

```
# setfacl -Rdm u:brais:rx /comunPRI #Con esta ACL aínda brais non ten acceso aos cartafolés /comunPRI/ana e /comunPRI/resto, xa que este permiso é para cando se xeneren ficheiros e directorios en eses cartafolés pero non para eses cartafolés xa existentes.
# setfacl -m u:brais:rx /comunPRI/ana #Con esta ACL agora si ten acceso a /comunPRI/ana
# setfacl -m u:brais:rx /comunPRI/resto #Con esta ACL agora si ten acceso a /comunPRI/resto
# setfacl -m u:brais:rx /comunPRI/ #Esta ACL non é necesaria porque brais xa ten acceso a /comunPRI por pertencer ao grupo secundaria
# getfacl -R /comunPRI
```

G. Eliminar todas as ACLs de /comunPRI

```
# setfacl -Rb /comunPRI
# getfacl -R /comunPRI
```

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/)

# Configuración de parámetros de rede

**ifconfig, ip, route, /etc/resolv.conf, dhclient**

- **ifconfig** (deprecated) (apt install net-tools): Permite configurar IP/Máscara das interfaces de rede.
- **ip**: Permite configurar IP/Máscara das interfaces de rede.
- **route** (deprecated) (apt install net-tools): Permite configurar o enrutamento/portas enlace(gateway).
- **/etc/resolv.conf**: Arquivo onde se configuran os servidores DNS.
- **dhclient**: Configuración dinámica da rede mediante solicitude a un servidor DHCP.

## Práctica: ifconfig e ip

```
$ ifconfig #Listar interfaces activas. Normalmente como usuario sen permisos de root (sudo) dá erro.

$ /sbin/ifconfig #Listar interfaces activas

$ /sbin/ifconfig -a #Listar interfaces estén ou non activas

# ifconfig #Listar interfaces activas

# ifconfig -a #Listar interfaces estén ou non activas

# ifconfig eth0 #Listar a configuración da interface eth0

# ifconfig eth0 up #Activar interface eth0

# ifconfig eth0 down #Deshabilitar interface eth0

# ifconfig eth0 192.168.100.100 #Configuración de rede para a interface eth0: IP=192.168.100.100,
MS=255.255.255.0

# ifconfig eth0 192.168.100.100/24 #Configuración de rede para a interface eth0: IP=192.168.100.100,
MS=255.255.255.0

# ifconfig eth0 192.168.100.100 netmask 255.255.255.0 #Equivale ao comando anterior.

# ifconfig eth0:0 192.168.100.101/24 #Xerar o alias eth0:0 para a interface eth0 con outra configuración de rede:
IP=192.168.100.101, MS=255.255.255.0.

# ifconfig eth0:0 192.168.100.101 netmask 255.255.255.0 #Equivale ao comando anterior.

# ifconfig eth0:web 192.168.100.102 netmask 255.255.255.0 #Novo alias eth0:web para a interface eth0.

$ ip addr #Listar interfaces activas

$ ip addr show #Equivale ao comando anterior.

$ ip addr show eth0 #Listar a configuración da interface eth0

# ip addr show #Listar interfaces activas

# ip addr show eth0 #Listar a configuración da interface eth0

# ip link set eth0 up #Activar interface eth0

# ip link set eth0 down #Deshabilitar interface eth0

# ip address add 192.168.100.100 dev eth0 #Configuración de rede para a interface eth0: IP=192.168.100.100,
MS=255.255.255.255. NOTA: Se eth0 coa IP 192.168.100.100 xa fora configurada con ifconfig, a nova configuración
realizada mediante ip non se vería na execución de ifconfig.

# ip address add 192.168.100.100/24 broadcast 192.168.100.255 dev eth0 #Configuración de rede para a
interface eth0: IP=192.168.100.100, MS=255.255.255.0

# ip address add 192.168.100.104/24 dev eth0 label eth0:4 #Xerar o alias eth0:4 para a interface eth0 con outra
configuración de rede: IP=192.168.100.104, MS=255.255.255.0.

# ip address del 192.168.100.101/24 dev eth0 #Eliminar esa configuración IP na interface eth0. Neste caso esa
configuración corresponden co alias eth0:0, polo que elimina ese alias da configuración de rede.

# ip address del 192.168.100.102/24 dev eth0:web #Eliminar o alias eth0:web
```

## Práctica: route e ip route

```
# route #Listar táboa de enrutamento.

# ip route #Listar táboa de enrutamento

# ip route show #Equivale ao comando anterior.

# ip route list #Equivale ao comando anterior.

# route -n #Listar táboa de enrutamento sen resolución DNS

# route add default gw 192.168.100.1 #Configurar porta de enlace (gateway).

# ip route add default via 192.168.100.1 #Equivale ao comando anterior.

# route del default gw 192.168.100.1 #Eliminar porta de enlace (gateway).

# ip route del default via 192.168.100.1 #Equivale ao comando anterior.

# route add -net 192.168.200.0 netmask 255.255.255.0 dev eth0 #Engadir regra de enrutamento para a rede 192.168.200.0 na interface eth0

# ip route add 192.168.200.0/24 dev eth0 #Equivale ao comando anterior.

# route del -net 192.168.200.0 netmask 255.255.255.0 dev eth0 #Eliminar regra de enrutamento para a rede 192.168.200.0 na interface eth0

# ip route delete 192.168.200.0/24 dev eth0 #Equivale ao comando anterior.

# route add -net 192.168.100.0 netmask 255.255.255.0 gw 192.168.100.1 dev eth0 #Engadir regra de enrutamento para a rede 192.168.100.0 na interface eth0 sendo a porta de enlace: 192.168.100.1

# ip route add 192.168.100.0/24 via 192.168.100.1 dev eth0 onlink #Equivale ao comando anterior.
```

## Práctica: /etc/resolv.conf

```
# echo 'domain example.local' > /etc/resolv.conf #Dominio a engadir na procura de hostnames. Se o host a buscar é pepito, é a procura falla, intentariase de novo esta como pepito.example.local

# echo 'search example.local' > /etc/resolv.conf #Lista de dominios a engadir na procura de hostnames.

domain e search son excluintes, a última directiva que apareza no ficheiro prevalece.

# echo 'nameserver 8.8.8.8' >> /etc/resolv.conf #Agregar servidor DNS primario para resolución de nomes.

# echo 'nameserver 8.8.4.4' >> /etc/resolv.conf #Agregar servidor DNS secundario para resolución de nomes.
```

## Práctica: dhclient

```
# dhclient -v eth0 #Configuración dinámica de rede da interface eth0 en modo verbose(detallado).

# dhclient -s 192.168.200.254 -v eth0 #Configuración dinámica de rede da interface eth0 en modo verbose(detallado), procurando a configuración no servidor DHCP 192.168.200.254
```

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/)

# REGEXP (Expresións Regulares: ^, \$, {1}...)

## Básicas

REGEXP	..... VALOR
.	..... calquera character
[ ]	..... calquera dos caracteres entre corchetes. Por exemplo: [rfc] pode ser r, f ou c, é dicir, devolverá calquera cadea que posúa r, f ou c [a-z] expresa o rango das letras minúsculas, é dicir, poder ser calquera letra minúscula. [A-Z] pode ser calquera letra maiúscula. [a-Z] pode ser calquera letra minúscula ou maiúscula.
[ ^ ]	..... calquera character que non e/sté entre corchetes. Por exemplo: [^rfc] pode ser calquera character agás r, f ou c [^a-z] calquera character que non sexa unha letra minúscula [^A-Z] calquera character que non sexa unha letra maiúscula
^	..... inicio de liña
\$	..... final de liña
*	..... 0 o máis coincidencias da expresión regular anterior Por exemplo: [rfc]* pode ser cero ou calquera número de aparicións de r, f ou c, como: rrrr, ou cc.
\ ( \)	..... permite agrupar expresións regulares. Por exemplo: \(rfc\) pode ser o agrupamento de caracteres rfc. \(r.c\) pode ser calquera agrupamento de caracteres onde: o primeiro character sexa r, o segundo sexa calquera e o terceiro sexa c. \(r.c\) * pode ser cero ou máis coincidencias de calquera agrupamento de caracteres onde: o primeiro character sexa r, o segundo sexa calquera e o terceiro sexa c.
\	..... escapa un metacarácter para que non sexa interpretado como expresión regular. Por exemplo: \.tmp poder ser .tmp. Se non se escapara: .tmp pode ser calquera character que logo continúe con tmp, como: Atmp, 7tmp, etc.
Repeticións	
\{n\}	..... exactamente n coincidencias da expresión regular previa. Por exemplo: [a-z]\{2\} significa exactamente 2 coincidencias de calquera character minúscula
\{n,\}	..... como mínimo n coincidencias da expresión regular previa. Por exemplo: [a-z]\{2,\} significa como mínimo 2 coincidencias de calquera character minúscula
\{n,m\}	..... entre n e m coincidencias da expresión regular previa. Por exemplo: [a-z]\{2,4\} significa entre 2 e 4 coincidencias de calquera character minúscula

## Extendidas

REGEXP	..... VALOR
+	..... 1 ou máis coincidencias da expresión regular anterior. Por exemplo: [a-z]+ significa 1 ou máis coincidencias de calquera character minúscula
?	..... 0 ou 1 coincidencias da expresión regular anterior. Por exemplo: [a-z]? significa 0 ou 1 coincidencia de calquera character minúscula
\ ( \)	..... reemprázase por ( )
\{ \}	..... reemprázase por { }
[ :lower:]	..... caracteres alfabéticos minúsculas. Equivale a [a-z]
[ :upper:]	..... caracteres alfabéticos maiúsculas. Equivale a [A-Z]
[ :alpha:]	..... caracteres alfabéticos. Equivale a [A-Za-z] e tamén a [:lower:] + [:upper:]
[ :digit:]	..... caracteres numéricos. Equivale a [0-9]
[ :xdigit:]	..... caracteres hexadecimais. Equivale a [0-9A-Fa-f]
[ :alnum:]	..... caracteres alfanuméricos. Equivale a [0-9A-Za-z] e tamén a [:alpha:] + [:digit:]
[ :blank:]	..... caracteres espazo e tabulado
[ :cntrl:]	..... caracteres de control
[ :punct:]	..... caracteres de puntuación. Equivale a ! " # \$ % & ' ( ) * + , - . / : ; < = > ? @ [ \ ] ^ _ ` {   } ~
[ :graph:]	..... caracteres gráficos. Equivale a [:alnum:] + [:punct:]
[ :print:]	..... caracteres imprimibles. Equivale a [:alnum:] + [:punct:] + espazo
[ :space:]	..... caracteres de espazo. Equivale a: tabulado, nova liña, tabulado vertical, salto de páxina, retorno de carro e espazo



## NOTAS

- Os comandos `grep` e `sed` por defecto permiten as expresións regulares básicas, pero para poder empregar as extendidas:
  - En `grep` débese utilizar o parámetro `-E` ou ben o comando `egrep`.
  - En `sed` débese empregar o parámetro `-r`
- Etiquetado: As expresións regulares que se poñen entre `()` quedan etiquetadas, polo que se poden referenciar mediante `\n`, sendo `n` o número da etiqueta. Así:  
`([0-9])\.` coincide con calquera número que apareza antes do carácter punto (este número como vai agrupado e sendo a primeira agrupación, poderá ser chamado mediante `\1`) e logo do punto con mesmo número co primeiro carácter, como: `1.1`, `2.2`, `3.3`, etc pero non `1.2`, `4.5`, etc  
`([0-9])\1ABC` coincide con `11ABC` pero non con `1ABC`  
`([0-9])\.` coincide con `1.1.ABC` pero non con `1.ABC`  
`([0-9])\1ABC-([0-9])\2ABC` coincide con `7.7.ABC-4.4.ABC` pero non con `7.7ABC-4.4ABC`
- Mediante o comando `egrep 'REGEXP'`, ou `egrep -o 'REGEXP'` para que soamente devolva o patrón buscado, pódense probar as expresións regulares. Así:

```
$ egrep '[rfc]' #Ao premer <Enter> queda a espera da introdución dalgunha expresión para a súa validación, de tal éxito,
que se a expresión é válida esta repítese nunha nova liña, senón segue a espera dunha nova expresión para validar.
aarrbbffccdd
aarrbbffccdd
^C
$ egrep -o '[rfc]' #Ao premer <Enter> queda a espera da introdución dalgunha expresión para a súa validación, de tal
éxito, que se a expresión é válida soamente repite o patrón coincidente buscado nunha nova liña, senón segue a espera dunha nova
expresión para validar.
aarrbbffccdd
r
r
f
f
c
c
^C
$ egrep -o '[^rfc]' #Ao premer <Enter> queda a espera da introdución dalgunha expresión para a súa validación, de tal
éxito, que se a expresión é válida soamente repite o patrón coincidente buscado nunha nova liña, senón segue a espera dunha nova
expresión para validar.
aarrbbffccdd
a
a
b
b
d
d
^C
```

## Práctica

```
$ grep '^root' /etc/passwd #Lista todas as liñas do ficheiro /etc/passwd que comecen por root

$ grep '/bin/bash$' /etc/passwd #Lista todas as liñas do ficheiro /etc/passwd que rematen por /bin/bash

$ grep '/bin/false$' /etc/passwd #Lista todas as liñas do ficheiro /etc/passwd que rematen por /bin/false

$ sed -e 's/^/# /g' /tmp/file.txt #Modifica, no ficheiro /tmp/file.txt, todas as liñas que comecen por un espazo para que
comecen co carácter '#'

$ sed -e 's/^$/# /g' /tmp/file.txt #Modifica, no ficheiro /tmp/file.txt, todas as liñas en branco polo carácter '#'

$ grep -v '^#' /tmp/file.sh #Lista todas as liñas do script /tmp/file.sh que non comecen polo carácter '#', e dicir, lista todas as
liñas que non sexan comentarios.

$ sed '/^$/d' /tmp/file.txt #Lista todas as liñas do ficheiro /tmp/file.txt agás as liñas en branco, pero non as elimina do ficheiro

$ sed -i '/^$/d' /tmp/file.txt #Elimina todas as liñas en branco do ficheiro /tmp/file.txt

$ ip addr | egrep -o '[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}' #Lista IP das interfaces de rede

$ ip addr | grep -Eo '[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}' #Equivale ao comando anterior

$ ip addr | grep -Eo '([0-9]{1,3}\.)+[0-9]{1,3}' #Equivale ao comando anterior

$ ip addr | grep -Eo '([0-9]{1,3}\.)+[0-9]{1,3}' | egrep -v '^127|255$' #Equivale ao comando anterior, pero non amosa
IP que comecen por 127 ou rematen por 255
```

```
$ /sbin/ifconfig | egrep -o '([0-9]{1,3})\.[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}' | egrep -v '255$|0$|1$' #Equivale ao comando anterior, pero non amosa IP que rematen por 255 ou 0
```

```
$ /sbin/ifconfig | egrep -o '([0-9a-f]{2}):([0-9a-f]{2}):([0-9a-f]{2}):([0-9a-f]{2}):([0-9a-f]{2}):([0-9a-f]{2})' #Lista MAC das interfaces de rede
```

```
$ /sbin/ifconfig | egrep -o '([0-9a-f]{2}):([0-9a-f]{2}):([0-9a-f]{2}):([0-9a-f]{2}):([0-9a-f]{2}):([0-9a-f]{2})' #Equivale ao comando anterior
```

```
$ /sbin/ifconfig | egrep -o '([0-9a-f]{2}):([0-9a-f]{2}):([0-9a-f]{2}):([0-9a-f]{2}):([0-9a-f]{2}):([0-9a-f]{2})' #Equivale ao comando anterior
```

```
$ /sbin/ifconfig | egrep -o '([0-9a-f]{2}):([0-9a-f]{2}):([0-9a-f]{2}):([0-9a-f]{2}):([0-9a-f]{2}):([0-9a-f]{2})' #Equivale ao comando anterior
```

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/)

# apropos, whatis, whereis, which, updatedb, locate

## Práctica

# **apropos nome** #Buscar en que páxinas do man existen referencias ao nome dado

# **apropos halt** #Buscar en que páxinas do man existen referencias ao comando halt

halt (5) - variables that affect the behavior of the shutdown scripts

halt (8) - Halt, power-off or reboot the machine

poweroff (8) - Halt, power-off or reboot the machine

reboot (8) - Halt, power-off or reboot the machine

shutdown (8) - Halt, power-off or reboot the machine

systemd-halt.service (8) - System shutdown logic

# **apropos -r nome** #Buscar en que páxinas do man existen referencias á expresion regular nome dada

# **apropos -r '^user\$'** #Buscar en que páxinas do man existen referencias á expresion regular '^user\$'.

arpd (8) - userspace arp daemon.

environ (7) - user environment

finger (1) - user information lookup program

group (5) - user group file

netrc (5) - user configuration for ftp

telnet (1) - user interface to the TELNET protocol

telnet.netkit (1) - user interface to the TELNET protocol

udp (7) - User Datagram Protocol for IPv4

user-dirs.conf (5) - configuration for xdg-user-dirs-update

user-dirs.defaults (5) - default settings for XDG user dirs

user-dirs.dirs (5) - settings for XDG user dirs

user\_clusters (5) - File linking users to PostgreSQL clusters

user\_namespaces (7) - overview of Linux user\_namespaces

useradd (8) - create a new user or update default new user information

userdel (8) - delete a user account and related files

usermod (8) - modify a user account

users (1) - print the user names of users currently logged in to the current host

xset (1) - user preference utility for X

# **whatis nome** #Buscar entre os nomes das páxinas do man, amosando aquelas que se asemellen ao nome dado

# **whatis halt** #Buscar entre os nomes das páxinas do man, amosando aquelas que se asemellen ao comando halt

halt (8) - Halt, power-off or reboot the machine

halt (5) - variables that affect the behavior of the shutdown scripts

# **whatis -r nome** #Buscar entre os nomes das páxinas do man, amosando aquelas que se asemellen á expresion regular nome dada

# **whatis -r '^user'** #Buscar entre os nomes das páxinas do man, amosando aquelas que se asemellen á expresion regular '^user'

user-dirs.conf (5) - configuration for xdg-user-dirs-update  
user-dirs.defaults (5) - default settings for XDG user dirs  
user-dirs.dirs (5) - settings for XDG user dirs  
user\_clusters (5) - File linking users to PostgreSQL clusters  
user\_namespaces (7) - overview of Linux user\_namespaces  
useradd (8) - create a new user or update default new user information  
userdel (8) - delete a user account and related files  
usermod (8) - modify a user account  
users (1) - print the user names of users currently logged in to the current host

# whereis nome #Buscar binarios, fontes e páxinas do man para o nome dado

# whereis passwd #Buscar binarios, fontes e páxinas do man para o comando passwd

passwd: /usr/bin/passwd /etc/passwd /usr/share/man/man5/passwd.5.gz /usr/share/man/man1/passwd.1ssl.gz  
/usr/share/man/man1/passwd.1.gz

# whereis -b passwd #Buscar binarios para o comando passwd

passwd: /usr/bin/passwd /etc/passwd

# whereis -s passwd #Buscar fontes do man para o comando passwd

passwd:

# whereis -m passwd #Buscar páxinas do man para o comando passwd

passwd: /usr/share/man/man5/passwd.5.gz /usr/share/man/man1/passwd.1ssl.gz /usr/share/man/man1/passwd.1.gz

# which nome #Buscar no PATH o nome dado, devolvendo se existe a ruta atopada

# which passwd #Buscar no PATH o comando passwd

/usr/bin/passwd

# updatedb #Crear ou actualizar unha base de datos usada polo comando locate

# locate nome #Buscar ficheiros según o nome dado na base de datos xerada por updatedb

# locate usermod #Buscar ficheiros referentes a usermod na base de datos xerada por updatedb

/home/usr/sbin/usermod

/home/usr/share/bash-completion/completions/usermod

/home/usr/share/bash-completion/completions/usermod

/home/usr/share/man/de/man8/usermod.8.gz

/home/usr/share/man/fr/man8/usermod.8.gz

/home/usr/share/man/it/man8/usermod.8.gz

/home/usr/share/man/ja/man8/usermod.8.gz

/home/usr/share/man/man8/usermod.8.gz

/home/usr/share/man/pl/man8/usermod.8.gz

/home/usr/share/man/ru/man8/usermod.8.gz

/home/usr/share/man/tr/man8/usermod.8.gz

/home/usr/share/man/zh\_CN/man8/usermod.8.gz

/home/usr/share/man/zh\_TW/man8/usermod.8.gz

/usr/sbin/usermod

```
/usr/share/bash-completion/completions/lusermod
/usr/share/bash-completion/completions/usermod
/usr/share/man/de/man8/usermod.8.gz
/usr/share/man/fr/man8/usermod.8.gz
/usr/share/man/it/man8/usermod.8.gz
/usr/share/man/ja/man8/usermod.8.gz
/usr/share/man/man8/usermod.8.gz
/usr/share/man/pl/man8/usermod.8.gz
/usr/share/man/ru/man8/usermod.8.gz
/usr/share/man/tr/man8/usermod.8.gz
/usr/share/man/zh_CN/man8/usermod.8.gz
/usr/share/man/zh_TW/man8/usermod.8.gz
/var/cache/man/cat8/usermod.8.gz
```

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/)

# halt, poweroff, reboot, shutdown

**halt, poweroff, reboot** posúen a mesma páxina do manual, empregando as mesmas opcións co mesmo significado. Así, opcións:

- f, --force: Forzar sen contactar con init**
- halt: Equivale a halt (deter)**
- p, --poweroff: Equivale a poweroff (apagar). Non funciona co comando reboot**
- reboot: Equivale a reboot (reiniciar)**

**shutdown** pecha o sistema de modo seguro. Todos os usuarios son notificados que o sistema estase a pechar e login é bloqueado.

- H, --halt: Equivale a halt**
- h: Equivale a poweroff a non ser que --halt sexa especificado**
- P: Equivale a poweroff**
- r: Equivale a reboot**

En Debian 8 (Jessie):

```
# which halt poweroff reboot shutdown | xargs -l {} ls -alh {}  
lrwxrwxrwx 1 root root 14 dic 19 09:08 /sbin/halt -> /bin/systemctl  
lrwxrwxrwx 1 root root 14 dic 19 09:08 /sbin/poweroff -> /bin/systemctl  
lrwxrwxrwx 1 root root 14 dic 19 09:08 /sbin/reboot -> /bin/systemctl  
lrwxrwxrwx 1 root root 14 dic 19 09:08 /sbin/shutdown -> /bin/systemctl
```

## Práctica

# halt #Apagar, deter ou reiniciar o sistema. Por defecto (Sen opcións), detén o sistema despois de pechalo invocando o comando shutdown -h.

# poweroff #Apagar, deter ou reiniciar o sistema. Por defecto (Sen opcións), apaga o sistema.

# reboot #Apagar, deter ou reiniciar o sistema. Por defecto (Sen opcións), reinicia o sistema despois de pechalo invocando o comando shutdown -r.

# halt -p #Equivale a poweroff

# halt --reboot #Equivale a reboot

# halt -f #Forza o apagado sen contactar con init

# poweroff --halt #Equivale a halt

# poweroff --reboot #Equivale a reboot

# poweroff -f #Forza o apagado do sistema sen contactar con init



# reboot --halt #Equivale a halt

# reboot -p #Equivale a **reboot**

# reboot -f #Forza o reinicio sen contactar con init

# shutdown #Apagar, deter ou reiniciar o sistema. Sen opcións programa o apagado do sistema no próximo minuto.

# shutdown -h time #Apagar o sistema á hora determinada en time, que pode vir dado en formato "hh:mm" horas/minutos (formato 24h), "+m" (+minutos da data presente), "now" agora (alias de "+0").

Cando time é empregado 5 minutos antes do apagado non se permite máis conexións ao sistema creando o ficheiro /run/nologin

# shutdown -h now #Apagar o sistema agora.

# shutdown -h 19:33 #Apagar o sistema ás 19:33

# shutdown -h time -k mensaxe #Apagar o sistema á hora determinada en time e amosa unha mensaxe aos usuarios conectados no momento da execución do comando determinado por time

# shutdown -h 19:33 -k 'Vaise apagar o sistema. Desconecte sesión.' #Apagar o sistema ás 19:33 amosando a mensaxe 'Vaise apagar o sistema. Desconecte sesión.'

# shutdown -c #Cancelar o apagado do sistema.

# shutdown -c -k mensaxe #Cancelar o apagado do sistema amosando unha mensaxe.

# shutdown -c -k 'Apagado cancelado' #Cancelar o apagado do sistema amosando a todos os usuarios conectados no momento da execución do comando a mensaxe 'Apagado cancelado'.

Ricardo Feijoo Costa



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

# shutdown e systemd

**shutdown** pecha o sistema de modo seguro. Todos os usuarios son notificados que o sistema estase a pechar e login é bloqueado.

**-H, --halt:** Equivale a halt

**-h:** Equivale a poweroff a non ser que --halt sexa especificado

**-P:** Equivale a poweroff

**-r:** Equivale a reboot

En Debian 8 (Jessie):

```
# which halt poweroff reboot shutdown | xargs -l {} ls -alh {}
```

```
lrwxrwxrwx 1 root root 14 dic 19 09:08 /sbin/halt -> /bin/systemctl
```

```
lrwxrwxrwx 1 root root 14 dic 19 09:08 /sbin/poweroff -> /bin/systemctl
```

```
lrwxrwxrwx 1 root root 14 dic 19 09:08 /sbin/reboot -> /bin/systemctl
```

```
lrwxrwxrwx 1 root root 14 dic 19 09:08 /sbin/shutdown -> /bin/systemctl
```

Cada comando shutdown executado sobreescribe o ficheiro **/run/systemd/shutdown/scheduled**, sendo o último comando shutdown executado o que prevalece. O ficheiro anterior ten unha sintaxe similar á seguinte:

```
USEC=1454283866132465
WARN_WALL=1
MODE=poweroff
DRY_RUN=1
WALL_MESSAGE=Apagando...

onde,
USEC e o tempo para a execución do shutdown en
microsegundos dende 1970-01-01 UTC
```

Para converter os segundos dende **1970-01-01 UTC** á data actual, empregamos o comando similar a:

```
$ date --date='@2147483647'
```

Entón para converter os microsegundos poderíamos executar o seguinte comando:

```
$ date --date=@$((1454283866132465/1000000))
```

```
lun feb 1 00:44:26 CET 2016
```

O comando **shutdown** é controlado polos servizos **systemd-shutdown.service** e **systemd-shutdown.socket**, co cal podemos empregar con estes servizos o comando **systemctl**.

## Práctica

**# shutdown** #Apagar, deter ou reiniciar o sistema. Sen opcións programa o apagado do sistema no próximo minuto.

**# shutdown -h time** #Apagar o sistema á hora determinada en time, que pode vir dado en formato "hh:mm" horas/minutos (formato 24h), "+m" (+minutos da data presente), "now" agora (alias de "+0").

Cando time é empregado 5 minutos antes do apagado non se permite máis conexións ao sistema creando o ficheiro **/run/nologin**

# shutdown -h now #Apagar o sistema agora.

# shutdown -h 19:33 #Apagar o sistema ás 19:33

# shutdown -h time -k mensaxe #Apagar o sistema á hora determinada en time e amosa unha mensaxe aos usuarios conectados no momento da execución do comando determinado por time

# shutdown -h 19:33 -k 'Vaise apagar o sistema. Desconecte sesión.' #Apagar o sistema ás 19:33 amosando a mensaxe 'Vaise apagar o sistema. Desconecte sesión.'

# shutdown -c #Cancelar o apagado do sistema.

# shutdown -c -k mensaxe #Cancelar o apagado do sistema amosando unha mensaxe.

# shutdown -c -k 'Apagado cancelado' #Cancelar o apagado do sistema amosando a todos os usuarios conectados no momento da execución do comando a mensaxe 'Apagado cancelado'.

## Práctica

# shutdown -r +1 #Reiniciar o sistema dentro de 1 minuto. A tarefa programada do shutdown gárdase no arquivo /run/systemd/shutdown/scheduled

Shutdown scheduled for lun 2016-02-01 00:25:26 CET, use 'shutdown -c' to cancel.

# cat /run/systemd/shutdown/scheduled

USEC=1454282726493892

WARN\_WALL=1

MODE=reboot

# shutdown -r +15 #Reiniciar o sistema dentro de 15 minutos. A tarefa programada do shutdown substitúe á anterior modificando o arquivo /run/systemd/shutdown/scheduled

Shutdown scheduled for lun 2016-02-01 00:39:31 CET, use 'shutdown -c' to cancel.

# cat /run/systemd/shutdown/scheduled

USEC=1454283571383077

WARN\_WALL=1

MODE=reboot

# shutdown -r +150 #Reiniciar o sistema dentro de 150 minutos. A tarefa programada do shutdown substitúe á anterior modificando o arquivo /run/systemd/shutdown/scheduled

Shutdown scheduled for lun 2016-02-01 02:54:38 CET, use 'shutdown -c' to cancel.

# cat /run/systemd/shutdown/scheduled

USEC=1454291678108695

WARN\_WALL=1

MODE=reboot

```
# shutdown -h +150 #Apagar o sistema dentro de 150 minutos. A tarefa programada do shutdown substitúe á anterior modificando o arquivo /run/systemd/shutdown/scheduled
```

Shutdown scheduled for lun 2016-02-01 03:00:43 CET, use 'shutdown -c' to cancel.

```
# cat /run/systemd/shutdown/scheduled
```

USEC=1454292043597369

WARN\_WALL=1

MODE=poweroff

```
# shutdown -h +10 -k 'Apagando...' #Apagar o sistema dentro de 10 minutos amosando a mensaxe 'Apagando...'. A tarefa programada do shutdown substitúe á anterior modificando o arquivo /run/systemd/shutdown/scheduled
```

Shutdown scheduled for lun 2016-02-01 03:00:43 CET, use 'shutdown -c' to cancel.

```
# cat /run/systemd/shutdown/scheduled
```

USEC=1454283866132465

WARN\_WALL=1

MODE=poweroff

DRY\_RUN=1

WALL\_MESSAGE=Apagando...

```
# shutdown -c #Cancelar shutdown. Ainda que houbo múltiples shutdown consecutivos lanzados, soamente o último é o que prevalece, de tal xeito que soamente é necesario lanzar unha vez o comando shutdown -c
```

```
# cat /run/systemd/shutdown/scheduled
```

cat: /run/systemd/shutdown/scheduled: No existe el fichero o el directorio

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/)

# Captura de teclas: Comando bind

## Práctica

¿Cómo capturar teclas en scripts e darlles un comando? Mediante o comando **bind** podemos capturar combinacións de teclas. Previamente á captura de combinacións de teclas deberíamos comprobar cales existen na shell Bash mediante o comando:

```
bind -p
```

Os arquivos **/etc/inputrc** e **\$HOME/.inputrc** conteñen definicións das combinacións de teclas, como por exemplo: que é o que fan as frechas, avpág, repág... e son lidos pola shell Bash cando arranca. A sintaxe do comando **bind** para poder executar, mediante combinación de teclas, comandos é a seguinte:

```
bind '"\eCódigo_teclas":"comandos"'
```

onde, *Código\_teclas* son os códigos das teclas e *comandos* son os comandos a executar cando se empregue a tecla ou combinación de teclas. Para saber o código dunha tecla, facer o seguinte:

1. Executar o comando **read**
2. Premer tecla
3. Veremos o código da tecla pulsada, do cal interésanos todos os caracteres agás os 2 primeiros: **^[**

Por exemplo:

1. Executar o comando **read**
2. Premer F8
3. Veremos o código **^[19~**, do cal interésanos **[19~**

O seguinte exemplo intentarán clarificar o exposto:

### EXEMPLO A: Script comandos-bind.sh

```
#!/bin/bash
#set -o emacs #Para activar a edición de liña
bind '"\e[19~":"/sbin/ifconfig eth0\n"' #F8
bind '"\e[21~":"exit\n"' #F10
bind '"\ep":"tail -f /var/log/dmesg\n"' #ESCp
bind '"\C-h":"history | grep ssh\n"' #Ctrl+h
```

A execución do script debe facerse mediante a seguinte orde, para que esté activa na shell da execución:

```
source comandos-bind.sh
```

**NOTA:** Ter en conta que a execución do comando **bind** para a combinación de teclas coa tecla **Ctrl** cambia.

Se o que se quere é sempre ter estas combinacións de teclas, incorporalas no arquivo **\$HOME/.bashrc**

**Para máis información:** [man readline \(Key Bindings\)](#)

Ricardo Feijoo Costa



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

# (Des)Montar Sistemas de ficheiros

- **Montar** un sistema de ficheiros significa facer uso deste sistema nunha ruta (directorio) do sistema operativo.
- **Desmontar** un sistema de ficheiros significa deixar de facer uso deste sistema facéndoo non accesible a través dunha ruta (directorio) do sistema operativo.

## findmnt, mount, umount, /etc/fstab, /etc/mtab

- **findmnt**: Lista tódolos sistemas de ficheiros montados: destino de montaxe, orixe de montaxe, tipo de sistema de ficheiros e opcións de montaxe. Tamén pode buscar por un sistema de ficheiros pasándoo como parámetro. Busca a información en /etc/fstab, /etc/mtab ou /proc/self/mountinfo.

```
$ findmnt #Amosa todos os sistemas de ficheiros montados e as opcións de montaxe.
```

```
$ findmnt /dev/sda1 #Amosa información sobre a partición montada: a partición 1 do disco /dev/sda
```

- **mount**: Amosa tódolos sistemas de ficheiros montados: orixe de montaxe, destino de montaxe, sistema de ficheiros e opcións de montaxe. Tamén permite montar un sistema de ficheiros.

```
$ mount #Amosa todos os sistemas de ficheiros montados e as opcións de montaxe.
```

```
# mount -a #Monta de forma automática todos os sistemas de ficheiros existentes no arquivo /etc/fstab, agás aqueles que posúan a opción de montaxe noauto.
```

```
# mount -t ext4 /dev/sdc1 /mnt #Monta a partición 1 do disco /dev/sdc en /mnt indicando o formato da partición, é dicir, indicando o tipo de sistema de ficheiros a montar.
```

```
# mount -t auto /dev/sdc1 /mnt #Monta a partición 1 do disco /dev/sdc en /mnt sen indicar o formato da partición, é dicir, sen indicar o tipo de sistema de ficheiros a montar. Así, mount intenta averiguar o tipo de sistema de ficheiros a montar que non aparezan etiquetados como nodev: primeiro intenta con blkid, logo con /etc/filesystems e logo con /proc/filesystems
```

```
# mount /dev/sdc1 /mnt #Equivale ao comando anterior.
```

```
# mount /dev/sdc1 #Busca a información de montaxe en /etc/fstab, é dicir, busca se existe unha liña onde apareza /dev/sdc1 como directorio orixe de montaxe e emprega a información existente na mesma para realizar a montaxe.
```

```
# mount /mnt #Busca a información de montaxe en /etc/fstab, é dicir, busca se existe unha liña onde apareza /mnt como directorio destino de montaxe e emprega a información existente na mesma para realizar a montaxe.
```

```
# mount -o remount,ro /mnt #Remontar /mnt en modo só lectura. Así, busca a información de montaxe en /etc/fstab, é dicir, busca se existe unha liña onde apareza /mnt como directorio destino de montaxe e emprega a información existente na mesma para realizar a montaxe, pero neste caso coa opción ro (read only).
```

```
# mount -o remount,rw /mnt #Remontar /mnt en modo lectura e escritura. Así, busca a información de montaxe en /etc/fstab, é dicir, busca se existe unha liña onde apareza /mnt como directorio destino de montaxe e emprega a información existente na mesma para realizar a montaxe, pero neste caso coa opción rw (read write).
```

```
# mount --bind /proc /mnt #Remontar /proc en /mnt. Despois desta chamada /proc é accesible en 2 directorios: /proc e /mnt
```

- **umount**: Permite desmontar os sistemas de ficheiros montados.

```
# umount -a #Desmonta tódolos sistemas de ficheiros descritos en /proc/self/mountinfo, agás proc, devfs, devpts, sysfs, rpc_pipefs e nfsd.
```

```
# umount -t ext4 /dev/sdc1 /mnt #Desmonta a partición 1 do disco /dev/sdc en /mnt indicando o formato da partición, é dicir, indicando o tipo de sistema de ficheiros a desmontar.
```

```
# umount /dev/sdc1 /mnt #Equivale ao comando anterior.
```

```
# umount /mnt #Busca a información de montaxe en /etc/fstab, é dicir, busca se existe unha liña onde apareza /mnt como directorio destino de montaxe e emprega a información existente na mesma para realizar a desmontaxe.
```



# umount /dev/sdc1 #Busca a información de montaxe en /etc/fstab, é dicir, busca se existe unha liña onde apareza /dev/sdc1 como directorio orixe de montaxe e emprega a información existente na mesma para realizar a desmontaxe.

- /etc/fstab: Este ficheiro contén información estática sobre o sistema de ficheiros, a cal é cargada no arranque do sistema operativo. (Ver man fstab). Cada liña indica un sistema de ficheiros a montar. Normalmente só root ten permisos de montaxe, pero se nunha entrada (liña) aparece a opción de montaxe *user*, calquera usuario poderá montar o correspondente sistema de ficheiros.

# echo '/dev/sr0 /mnt iso9660 user 0 0' >> /etc/fstab #Engadir ao ficheiro /etc/fstab unha nova entrada para montar/desmontar o primeiro dispositivo CD/DVD: directorio orixe, directorio destino, tipo de sistema de ficheiros, opcións, dump 0(inexistente dump), fsck 0 (inexistente fsck)

\$ mount /dev/sr0 #Montar como usuario sen permisos de root o primeiro dispositivo CD/DVD.

\$ umount /dev/sr0 #Desmontar como usuario sen permisos de root o primeiro dispositivo CD/DVD.

\$ mount /mnt #Montar como usuario sen permisos de root o primeiro dispositivo CD/DVD.

\$ umount /mnt #Desmontar como usuario sen permisos de root o primeiro dispositivo CD/DVD.

- /etc/mtab: Este ficheiro garda a información en tempo real dos sistemas de ficheiros montados. Os comandos mount e umount manteñen actualizado este ficheiro. Hoxe en día soe ser unha ligazón simbólica a /proc/self/mounts (ou /proc/mounts).

### Sistemas de ficheiros virtuais: sysfs, proc, udev, tmpfs, run

- sysfs en /sys: Pseudo-sistema de ficheiros que proporciona unha interface á estrutura de datos do kernel (man 5 sysfs). Así proporciona información sobre dispositivos, módulos do kernel, sistemas de ficheiros e outros compoñentes do kernel. Soe ser montado automaticamente polo sistema operativo, pero pódese montar mediante o seguinte comando:

```
# mount -t sysfs sysfs /sys
```

- proc en /proc: Pseudo-sistema de ficheiros de información de procesos (man 5 proc)
- udev en /dev: Subministra o software do sistema con eventos de dispositivos, xestiona permisos de nodos e dispositivos e pode crear adicionais ligazóns no directorio /dev ou cambiar o nome de interfaces de rede.
- tmpfs en /tmp, en /run, en /run/lock, en /dev/shm: Sistema de ficheiros temporal que reside na memoria RAM ou na zona de intercambio (swap).

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/)

# chroot - Montaxe sistema de ficheiros



**ESCENARIO:** Para verificar o que acontece na práctica 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.
      - Táboa de particións msdos
      - Dúas particións primarias:
        - Raíz do sistema: /dev/sda1 (/). Formato: ext4
        - Swap: /dev/sda2 (swap). Formato: swap
      - 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



## Práctica chroot

1. Arrancar a máquina virtual 2 creada en modo Inicio normal.
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 arranca 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.
$ sudo su - #Acceder á consola de root(administrador) a través dos permisos configurados co comando sudo (/etc/sudoers, visudo)

# cat /proc/cmdline #Amosar o contido de /proc/cmdline que contén os parámetros cos cales o kernel foi arrancado.
# mount #Amosar os sistemas de ficheiros montados, é dicir, os que está a usar e podemos empregar neste sistema operativo live
debian.
# fdisk -l /dev/sda #Lista a táboa de particións do disco /dev/sda e logo remata.
# mkdir /mnt/recuperar #Crear o directorio /mnt/recuperar.
# mount -t auto /dev/sda1 /mnt/recuperar #Montar a partición 1 do disco duro /dev/sda no directorio da live /mnt/recuperar. Coa
opción -t auto solicitamos ao comando mount a autodetección do sistema de ficheiros de montaxe..
# mount --bind /dev /mnt/recuperar/dev # Montar o cartafol /dev dentro de /mnt/recuperar/dev para poder ter acceso a todos os
dispositivos recoñecidos pola distribución live.
# mount --bind /proc /mnt/recuperar/proc #Montar o cartafol /proc dentro de /mnt/recuperar/proc para poder ter acceso ao kernel
grazas a distribución live.
# mount --bind /sys /mnt/recuperar/sys #Montar o cartafol /sys dentro de /mnt/recuperar/sys para poder ter acceso ao kernel
grazas a distribución live.
# chroot /mnt/recuperar /bin/bash #Crear a xaula chroot. Con este comando creamos unha xaula: un entorno pechado para a
distribución Linux dentro de recuperar, de tal xeito, que unha vez dentro da xaula soamente existe ésta, e dicir, soamente existe a
distribución Linux instalada no disco duro /dev/sda a recuperar, xa non estamos traballando na Live.
# ping -c 4 www.google.es #Enviar 4 paquetes ICMP ECHO_REQUEST a www.google.es, solicitando 4 paquetes ICMP
ECHO_RESPONSE, para verificar a conectividade de rede hacia Internet e ao servidor de google.
# passwd usuario #Modificar o contraseñal do usuario de nome usuario. Pór como contraseñal 1234. Repetir o contraseñal. Olla:
Non aparecen asteriscos nin outro tipo de caracteres para impedir saber cantos e cales caracteres estamos a escribir.
# exit #Saír da xaula chroot para voltar á consola local do usuario root.
# umount /mnt/recuperar/dev /mnt/recuperar/proc /mnt/recuperar/sys /mnt/recuperar #Desmontar as unidades montadas.
# init 0 #Comando para enviar o runlevel (nivel de execución) do sistema operativo ao nivel 0, equivalente a apagar o sistema.
```
6. Pór como primeira opción de arranque o disco duro.
7. Arrancar a máquina virtual en modo Inicio normal e na opción por defecto do xestor de arranque GRUB.
8. Comprobar que agora o contraseñal do usuario root foi modificada.



# Sistemas de arquivos: Inodos

NOTA: Para verificar o que acontece na práctica crear unha máquina virtual en VirtualBox que arranque cunha ISO Live Debian 32bits, escritorio XFCE, 512MB de RAM e disco duro dinámico de 8GB. Imos supor que esta máquina virtual posúe o nome **Debian32-Inodos** e o disco duro posúe o nome **Debian32-Inodos.vdi**. Verificar que a primeira opción de arranque sexa o CD Virtual.



Por cada ficheiro ou directorio no sistema, existe un **inodo**, unha estrutura de datos, que garda a información do ficheiro, como por exemplo o seu propietario, permisos, marcas de tempos e localización dos seus bloques de datos no disco. É similar aos rexistros do MFT en NTFS.

**Sector** = **Bloque**

**Sector lóxico** != **Sector físico**



## Comandos de interese sobre sistemas de ficheiros ext2/ext3/ext4

- **ls** #O comando ls lista os ficheiros e os contidos de directorios.
- **df** #O comando df informa do emprego do espazo de disco en sistemas de ficheiros.
- **stat** #O comando stat permite amosar información sobre ficheiros ou sistemas de ficheiros.
- **debugfs** #O comando debugfs permite depurar sistemas de ficheiros ext2/ext3/ext4
- **dumpe2fs** #O comando dumpe2fs permite listar información sobre sistemas de ficheiros ext2/ext3/ext4.
- **tune2fs** #O comando tune2fs permite axustar os parámetros do sistema de ficheiros sobre sistemas de ficheiros ext2/ext3/ext4.

## debugfs

# **debugfs /dev/sda1** #Executar o comando debugfs sobre a partición primaria /dev/sda1, a cal non pode estar montada no sistema operativo, é dicir, non se pode estar facendo uso da partición para non corromper a depuración da partición a través do comando debugfs.

**debugfs: stat <X>** #Ver información sobre o inodo número X

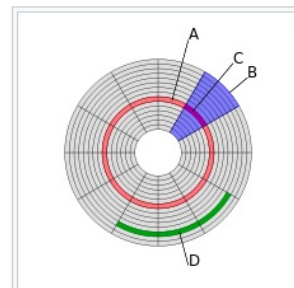
**debugfs: blocks <X>** #Ver os bloques que apuntan ao contido do ficheiro que corresponde co inodo X

**debugfs: cat <X>** #Ver o contido do ficheiro que corresponde co inodo X

**debugfs: ncheck X** #Identificar a relación entre o inodo X e o nome do arquivo ao que pertence.

**debugfs: ickcheck Y** #Identificar a relación entre o bloque Y e o inodo ao que pertence.

**debugfs: block\_dump Y** #Ver o contido do bloque Y. A información amósase en código hexadecimal e ASCII



Estructura de disco que muestra:

(A) una pista (roja),  
(B) un sector geométrico (azul),  
(C) un sector de una pista (magenta),  
(D) y un grupo de sectores o **clúster** (verde).

[Wikipedia](#)



1. Arrancar a máquina virtual creada en modo Inicio normal

`VBoxManage startvm Debian32-Inodos`

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

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

`$ sudo su - #Acceder á consola de root(administrador) a través dos permisos configurados co comando sudo (/etc/sudoers, visudo)`

**Crear e formatear particións**

`# parted --script /dev/sda mklabel msdos #Crear a etiqueta de disco (táboa de particións) ao dispositivo /dev/sda sen ter que acceder ao prompt de parted`

`# parted --script /dev/sda mkpart primary 0 50% -a cylinder #Crear unha partición primaria no disco /dev/sda cos primeiros 5GB, alineando a cilindros, sen ter que acceder ao prompt de parted`

`# parted --script /dev/sda mkpart primary 50% 70% -a cylinder #Crear unha partición primaria no disco /dev/sda de 2GB a continuación da partición de 5GB, alineando a cilindros, sen ter que acceder ao prompt de parted`

`# parted --script /dev/sda print #Amosa a táboa de particións do disco /dev/sda`

Model: ATA VBOX HARDDISK (scsi)  
Disk /dev/sda: 8590MB  
Sector size (logical/physical): 512B/512B  
Partition Table: msdos  
Disk Flags:

Number	Start	End	Size	Type	File system	Flags
1	32.3kB	4294MB	4294MB	primary		
2	4294MB	6013MB	1719MB	primary		

`# mkfs.ext4 -F -L 'PARTICION1' /dev/sda1 #Formatear en ext4 coa etiqueta PARTICION1 a partición primaria /dev/sda1`

mke2fs 1.43.4 (31-Jan-2017)  
/dev/sda1 contains a ext4 file system labelled 'PARTICION1'  
last mounted on /mnt/part1 on Mon Oct 29 17:58:50 2018  
Creating filesystem with 1048233 4k blocks and 262144 inodes  
Filesystem UUID: 459fb916-7189-4b3f-83de-fd81b56973f8  
Superblock backups stored on blocks:  
32768, 98304, 163840, 229376, 294912, 819200, 884736

Allocating group tables: done  
Writing inode tables: done  
Creating journal (16384 blocks): done  
Writing superblocks and filesystem accounting information: done

`# mkfs.ext3 -F -L 'PARTICION2' /dev/sda2 #Formatear en ext3 coa etiqueta PARTICION2 a partición primaria /dev/sda2`

mke2fs 1.43.4 (31-Jan-2017)  
/dev/sda2 contains a ext3 file system labelled 'PARTICION2'  
created on Mon Oct 29 17:58:29 2018  
Creating filesystem with 419698 4k blocks and 105040 inodes  
Filesystem UUID: d5c70817-fa3c-4734-8825-fa78463558cd  
Superblock backups stored on blocks:  
32768, 98304, 163840, 229376, 294912

Allocating group tables: done  
Writing inode tables: done  
Creating journal (8192 blocks): done  
Writing superblocks and filesystem accounting information: done

**Montar particións e crear ficheiros e directorios**

`# mkdir /mnt/part1 /mnt/part2 #Crear os directorios /mnt/part1 e /mnt/part2`

`# mount /dev/sda1 /mnt/part1 #Montar (facer uso) a partición primaria /dev/sda1 en /mnt/part1`

`# cd /mnt/part1 #Acceder ao directorio /mnt/part1`

`# echo 1 2 3 > 1.txt #Crear o ficheiro /mnt/part1/1.txt co contido 1 2 3`

`# mkdir -p cartafol1/cartafol2/cartafol3/cartafol4 #Crear a estrutura arbórea de directorios seguinte:`

```
/mnt/part1
├── 1.txt
├── cartafol1
│   ├── cartafol2
│   │   ├── cartafol3
│   │   │   └── cartafol4
└── lost+found
```

5 directories, 1 file

A opción -p permite crear os directorios pais que falten para cada argumento directorio

`# echo 4 5 6 > cartafol1/a.txt #Crear o ficheiro /mnt/part1/cartafol1/a.txt co contido 4 5 6`

`# echo 7 8 9 > cartafol1/cartafol2/b.txt #Crear o ficheiro /mnt/part1/cartafol1/cartafol2/b.txt co contido 7 8 9`

**Información sobre o emprego dos inodos nos sistemas de ficheiros**  
# df -i #Lista información sobre o emprego de inodos en lugar da información sobre os bloques.

Filesystem	Inodes	IUsed	IFree	IUse%	Mounted on
udev	125786	358	125428	1%	/dev
tmpfs	128624	487	128137	1%	/run
/dev/sr0	0	0	0	-	/lib/live/mount/medium
/dev/loop0	152322	152322	0	100%	/lib/live/mount/rootfs/filesystem.squashfs
tmpfs	128624	1	128623	1%	/lib/live/mount/overlay
overlay	128624	649	127975	1%	/
tmpfs	128624	1	128623	1%	/dev/shm
tmpfs	128624	3	128621	1%	/run/lock
tmpfs	128624	15	128609	1%	/sys/fs/cgroup
tmpfs	128624	19	128605	1%	/tmp
tmpfs	128624	19	128605	1%	/run/user/1000
/dev/sda1	262144	16	262128	1%	/mnt/part1
tmpfs	128624	11	128613	1%	/run/user/0
tmpfs	128624	14	128610	1%	/run/user/112

# df -hi #Lista información sobre o emprego de inodos en lugar da información sobre os bloques. A opción -h engade unha letra indicativa de tamaño, como M para megabytes binarios ('mebibytes'), a cada tamaño.

Filesystem	Inodes	IUsed	IFree	IUse%	Mounted on
udev	123K	358	123K	1%	/dev
tmpfs	126K	487	126K	1%	/run
/dev/sr0	0	0	0	-	/lib/live/mount/medium
/dev/loop0	149K	149K	0	100%	/lib/live/mount/rootfs/filesystem.squashfs
tmpfs	126K	1	126K	1%	/lib/live/mount/overlay
overlay	126K	649	125K	1%	/
tmpfs	126K	1	126K	1%	/dev/shm
tmpfs	126K	3	126K	1%	/run/lock
tmpfs	126K	15	126K	1%	/sys/fs/cgroup
tmpfs	126K	19	126K	1%	/tmp
tmpfs	126K	19	126K	1%	/run/user/1000
/dev/sda1	256K	16	256K	1%	/mnt/part1
tmpfs	126K	11	126K	1%	/run/user/0
tmpfs	126K	14	126K	1%	/run/user/112

## Listar/Ver información sobre inodos

```
# ls -l 1.txt #Listar de forma extendida o ficheiro /mnt/part1/1.txt

-rw-r--r-- 1 root root 6 Oct 29 19:57 1.txt

# ls -li 1.txt #Listar de forma extendida e amosando o inodo do ficheiro /mnt/part1/1.txt

12 -rw-r--r-- 1 root root 6 Oct 29 19:57 1.txt

# stat 1.txt #Amosar a información do comando stat sobre o ficheiro /mnt/part1/1.txt

  File: 1.txt
  Size: 6             Blocks: 8             IO Block: 4096   regular file
Device: 801h/2049d    Inode: 12             Links: 1
Access: (0644/-rw-r--r--)  Uid: (   0/   root)   Gid: (   0/   root)
Access: 2018-10-29 19:57:11.461449310 +0000
Modify: 2018-10-29 19:57:11.461449310 +0000
Change: 2018-10-29 19:57:11.461449310 +0000
 Birth: -

# ls -ld cartafol1 #Listar de forma extendida o directorio /mnt/part1/cartafol1

drwxr-xr-x 3 root root 4096 Oct 29 20:17 cartafol1

# ls -ldi cartafol1 #Listar de forma extendida e amosando o inodo do directorio /mnt/part1/cartafol1

131073 drwxr-xr-x 3 root root 4096 Oct 29 20:17 cartafol1

# ls -lR cartafol1 #Listar de forma extendida e recursiva o contido do directorio /mnt/part1/cartafol1

cartafol1:
total 8
-rw-r--r-- 1 root root    6 Oct 29 20:17 a.txt
drwxr-xr-x 3 root root 4096 Oct 29 20:17 cartafol2

cartafol1/cartafol2:
total 8
-rw-r--r-- 1 root root    6 Oct 29 20:17 b.txt
drwxr-xr-x 3 root root 4096 Oct 29 20:17 cartafol3

cartafol1/cartafol2/cartafol3:
total 4
drwxr-xr-x 2 root root 4096 Oct 29 20:17 cartafol4

cartafol1/cartafol2/cartafol3/cartafol4:
total 0

# ls -lRi cartafol1 #Listar de forma extendida e recursiva, amosando os inodos, do contido do directorio /mnt/part1/cartafol1

cartafol1:
total 8
131077 -rw-r--r-- 1 root root    6 Oct 29 20:17 a.txt
131074 drwxr-xr-x 3 root root 4096 Oct 29 20:17 cartafol2

cartafol1/cartafol2:
total 8
131078 -rw-r--r-- 1 root root    6 Oct 29 20:17 b.txt
131075 drwxr-xr-x 3 root root 4096 Oct 29 20:17 cartafol3

cartafol1/cartafol2/cartafol3:
total 4
131076 drwxr-xr-x 2 root root 4096 Oct 29 20:17 cartafol4

cartafol1/cartafol2/cartafol3/cartafol4:
total 0

# stat cartafol1 #Amosar a información do comando stat sobre o directorio /mnt/part1/cartafol1

  File: cartafol1
  Size: 4096          Blocks: 8             IO Block: 4096   directory
Device: 801h/2049d    Inode: 131073         Links: 3
Access: (0755/drwxr-xr-x)  Uid: (   0/   root)   Gid: (   0/   root)
Access: 2018-10-29 22:32:28.104085550 +0000
Modify: 2018-10-29 20:17:39.900917734 +0000
Change: 2018-10-29 20:17:39.900917734 +0000
 Birth: -
```

## Desmontar e ver información gardada nos inodos

```
# cd #Acceder ao directorio casa do usuario (/home/user)
```

```
# umount /mnt/part1 #Desmontar (deixar de facer uso) a partición primaria /dev/sda1 que estaba montada en /mnt/part1
```

```
# debugfs /dev/sda1 #Executar o comando debugfs sobre a partición primaria /dev/sda1
```

```
debugfs 1.43.4 (31-Jan-2017)
debugfs: stat <12> #Ver información sobre o inodo 12
Inode: 12   Type: regular   Mode: 0644   Flags: 0x80000
Generation: 388472943   Version: 0x00000000:00000001
User:      0   Group:      0   Project:    0   Size: 6
File ACL: 0   Directory ACL: 0
Links: 1   Blockcount: 8
Fragment: Address: 0   Number: 0   Size: 0
          ctime: 0x5bd76617:6e04a178 -- Mon Oct 29 19:57:11 2018
          atime: 0x5bd76617:6e04a178 -- Mon Oct 29 19:57:11 2018
          mtime: 0x5bd76617:6e04a178 -- Mon Oct 29 19:57:11 2018
          crtime: 0x5bd76617:6e04a178 -- Mon Oct 29 19:57:11 2018
Size of extra inode fields: 32
Inode checksum: 0x6451e928
EXTENTS:
(0):33281
(END)
```

Premer a tecla **q** para voltar á consola debugfs

```
debugfs: stat <131073> #Ver información sobre o inodo 131073
Inode: 131073   Type: directory   Mode: 0755   Flags: 0x80000
Generation: 1274141839   Version: 0x00000000:00000003
User:      0   Group:      0   Project:    0   Size: 4096
File ACL: 0   Directory ACL: 0
Links: 3   Blockcount: 8
Fragment: Address: 0   Number: 0   Size: 0
          ctime: 0x5bd76ae3:d6cba798 -- Mon Oct 29 20:17:39 2018
          atime: 0x5bd78a7c:18d0e0b8 -- Mon Oct 29 22:32:28 2018
          mtime: 0x5bd76ae3:d6cba798 -- Mon Oct 29 20:17:39 2018
          crtime: 0x5bd76ac7:ad264d1c -- Mon Oct 29 20:17:11 2018
Size of extra inode fields: 32
Inode checksum: 0x99381db4
EXTENTS:
(0):532512
(END)
```

Premer a tecla **q** para voltar á consola debugfs

```
debugfs: q #Premer de novo a tecla q para saír da consola debugfs
```

## Ver a información gardada nos inodos. Tamén os punteiros dos bloques de disco que almacenan o contido dos ficheiros e directorios

```
# mount /dev/sda1 /mnt/part1 #Montar (facer uso) a partición primaria /dev/sda1 en /mnt/part1

# cd /mnt/part1 #Acceder ao directorio /mnt/part1

# dd if=/dev/zero of=/mnt/part1/file0.txt bs=1MiB count=1 #Crear un ficheiro /mnt/part1/file1.txt que contén todos
ceros cun tamaño de 1MiB.

1+0 records in
1+0 records out
1048576 bytes (1.0 MB, 1.0 MiB) copied, 0.0111686 s, 93.9 MB/s

# ls -li file0.txt #Listar de forma extendida e amosando o inodo do ficheiro /mnt/part1/file0.txt

15 -rw-r--r-- 1 root root 1048576 Oct 30 17:19 file0.txt

# cd #Acceder ao directorio casa do usuario (/home/user)

# umount /mnt/part1 #Desmontar (deixar de facer uso) a partición primaria /dev/sda1 que estaba montada en /mnt/part1

# debugfs /dev/sda1 #Executar o comando debugfs sobre a partición primaria /dev/sda1

debugfs 1.43.4 (31-Jan-2017)
debugfs: stat <15>
Inode: 15   Type: regular   Mode: 0644   Flags: 0x80000
Generation: 4057643325   Version: 0x00000000:00000001
User:      0   Group:      0   Project:      0   Size: 1048576
File ACL: 0   Directory ACL: 0
Links: 1   Blockcount: 2048
Fragment: Address: 0   Number: 0   Size: 0
  ctime: 0x5bd892a4:8e297e28 -- Tue Oct 30 17:19:32 2018
  atime: 0x5bd892a4:8d3559b4 -- Tue Oct 30 17:19:32 2018
  mtime: 0x5bd892a4:8e297e28 -- Tue Oct 30 17:19:32 2018
  crtime: 0x5bd892a4:8d3559b4 -- Tue Oct 30 17:19:32 2018
Size of extra inode fields: 32
Inode checksum: 0x13ebaa75
EXTENTS:
(0-255):33536-33791
(END)
```

Premer a tecla **q** para voltar á consola debugfs

```
debugfs: blocks <15> #Ver os bloques que apuntan ao contido do ficheiro que corresponde co inodo 15

33536 33537 33538 33539 33540 33541 33542 33543 33544 33545 33546 33547 33548 33549 33550 33551 33552
33553 33554 33555 33556 33557 33558 33559 33560 33561 33562 33563 33564 33565 33566 33567 33568 33569
33570 33571 33572 33573 33574 33575 33576 33577 33578 33579 33580 33581 33582 33583 33584 33585 33586
33587 33588 33589 33590 33591 33592 33593 33594 33595 33596 33597 33598 33599 33600 33601 33602 33603
33604 33605 33606 33607 33608 33609 33610 33611 33612 33613 33614 33615 33616 33617 33618 33619 33620
33621 33622 33623 33624 33625 33626 33627 33628 33629 33630 33631 33632 33633 33634 33635 33636 33637
33638 33639 33640 33641 33642 33643 33644 33645 33646 33647 33648 33649 33650 33651 33652 33653 33654
33655 33656 33657 33658 33659 33660 33661 33662 33663 33664 33665 33666 33667 33668 33669 33670 33671
33672 33673 33674 33675 33676 33677 33678 33679 33680 33681 33682 33683 33684 33685 33686 33687 33688
33689 33690 33691 33692 33693 33694 33695 33696 33697 33698 33699 33700 33701 33702 33703 33704 33705
33706 33707 33708 33709 33710 33711 33712 33713 33714 33715 33716 33717 33718 33719 33720 33721 33722
33723 33724 33725 33726 33727 33728 33729 33730 33731 33732 33733 33734 33735 33736 33737 33738 33739
33740 33741 33742 33743 33744 33745 33746 33747 33748 33749 33750 33751 33752 33753 33754 33755 33756
33757 33758 33759 33760 33761 33762 33763 33764 33765 33766 33767 33768 33769 33770 33771 33772 33773
33774 33775 33776 33777 33778 33779 33780 33781 33782 33783 33784 33785 33786 33787 33788 33789 33790
33791

debugfs: q Premer de novo a tecla q para saír da consola debugfs
```

```
# mount /dev/sda1 /mnt/part1 #Montar (facer uso) a partición primaria /dev/sda1 en /mnt/part1

# cd /mnt/part1 #Acceder ao directorio /mnt/part1

# dd if=/dev/zero of=/mnt/part1/file1.txt bs=1MiB count=10 #Crear un ficheiro /mnt/part1/file1.txt que contén todos
ceros cun tamaño de 10MiB.

10+0 records in
10+0 records out
10485760 bytes (10 MB, 10 MiB) copied, 0.0851866 s, 123 MB/s

# ls -li file1.txt #Listar de forma extendida e amosando o inodo do ficheiro /mnt/part1/file1.txt

14 -rw-r--r-- 1 root root 10485760 Oct 30 17:09 file1.txt

# cd #Acceder ao directorio casa do usuario (/home/user)

# umount /mnt/part1 #Desmontar (deixar de facer uso) a partición primaria /dev/sda1 que estaba montada en /mnt/part1

# debugfs /dev/sda1 #Executar o comando debugfs sobre a partición primaria /dev/sda1

debugfs 1.43.4 (31-Jan-2017)
debugfs: stat <14> #Ver información sobre o inodo 14
Inode: 14   Type: regular   Mode: 0644   Flags: 0x80000
Generation: 2934060132   Version: 0x00000000:00000001
User:      0   Group:      0   Project:      0   Size: 10485760
File ACL: 0   Directory ACL: 0
Links: 1   Blockcount: 20480
Fragment: Address: 0   Number: 0   Size: 0
  ctime: 0x5bd89063:c089d3a4 -- Tue Oct 30 17:09:55 2018
  atime: 0x5bd88fa1:2c3b8298 -- Tue Oct 30 17:06:41 2018
  mtime: 0x5bd89063:c089d3a4 -- Tue Oct 30 17:09:55 2018
  crtime: 0x5bd88fa1:2c3b8298 -- Tue Oct 30 17:06:41 2018
Size of extra inode fields: 32
```



88008	88065	88066	88067	88068	88069	88070	88071	88072	88073	88074	88075	88076	88077	88078	88079	88080
88081	88082	88083	88084	88085	88086	88087	88088	88089	88090	88091	88092	88093	88094	88095	88096	88097
88098	88099	88100	88101	88102	88103	88104	88105	88106	88107	88108	88109	88110	88111	88112	88113	88114
88115	88116	88117	88118	88119	88120	88121	88122	88123	88124	88125	88126	88127	88128	88129	88130	88131
88132	88133	88134	88135	88136	88137	88138	88139	88140	88141	88142	88143	88144	88145	88146	88147	88148
88149	88150	88151	88152	88153	88154	88155	88156	88157	88158	88159	88160	88161	88162	88163	88164	88165
88166	88167	88168	88169	88170	88171	88172	88173	88174	88175	88176	88177	88178	88179	88180	88181	88182
88183	88184	88185	88186	88187	88188	88189	88190	88191	88192	88193	88194	88195	88196	88197	88198	88199
88200	88201	88202	88203	88204	88205	88206	88207	88208	88209	88210	88211	88212	88213	88214	88215	88216
88217	88218	88219	88220	88221	88222	88223	88224	88225	88226	88227	88228	88229	88230	88231	88232	88233
88234	88235	88236	88237	88238	88239	88240	88241	88242	88243	88244	88245	88246	88247	88248	88249	88250
88251	88252	88253	88254	88255	88256	88257	88258	88259	88260	88261	88262	88263	88264	88265	88266	88267
88268	88269	88270	88271	88272	88273	88274	88275	88276	88277	88278	88279	88280	88281	88282	88283	88284
88285	88286	88287	88288	88289	88290	88291	88292	88293	88294	88295	88296	88297	88298	88299	88300	88301
88302	88303	88304	88305	88306	88307	88308	88309	88310	88311	88312	88313	88314	88315	88316	88317	88318
88319	88320	88321	88322	88323	88324	88325	88326	88327	88328	88329	88330	88331	88332	88333	88334	88335
88336	88337	88338	88339	88340	88341	88342	88343	88344	88345	88346	88347	88348	88349	88350	88351	88352
88353	88354	88355	88356	88357	88358	88359	88360	88361	88362	88363	88364	88365	88366	88367	88368	88369
88370	88371	88372	88373	88374	88375	88376	88377	88378	88379	88380	88381	88382	88383	88384	88385	88386
88387	88388	88389	88390	88391	88392	88393	88394	88395	88396	88397	88398	88399	88400	88401	88402	88403
88404	88405	88406	88407	88408	88409	88410	88411	88412	88413	88414	88415	88416	88417	88418	88419	88420
88421	88422	88423	88424	88425	88426	88427	88428	88429	88430	88431	88432	88433	88434	88435	88436	88437
88438	88439	88440	88441	88442	88443	88444	88445	88446	88447	88448	88449	88450	88451	88452	88453	88454
88455	88456	88457	88458	88459	88460	88461	88462	88463								

```
debugfs: q Premer de novo a tecla q para sair da consola debugfs
```

```
# mount /dev/sda1 /mnt/part1 #Montar (facer uso) a partición primaria /dev/sda1 en /mnt/part1
```

```
# cd /mnt/part1 #Acceder ao directorio /mnt/part1
```

```
# echo 'Cada día sabemos máis e entendemos menos. Einstein' > proverbio1.txt #Crear un ficheiro
/mnt/part1/proverbio1.txt que contén unha frase.
```

```
# cat proverbio1.txt #Ver o contido do ficheiro /mnt/part1/proverbio1.txt
```

Cada día sabemos máis e entendemos menos. Einstein

```
# ls -li proverbio1.txt #Listar de forma extendida e amosando o inodo do ficheiro /mnt/part1/proverbio1.txt
```

```
16 -rw-r--r-- 1 root root 53 Oct 30 18:33 proverb101.txt
```

```
# stat proverbio1.txt #Amosar a información do comando stat sobre o ficheiro /mnt/part1/proverbio1.txt
```

```
File: proverb10.txt
Size: 53          Blocks: 8          IO Block: 4096    regular file
Device: 801h/2049d Inode: 16         Links: 1
Access: (0644/-rw-r--r--)  Uid: (    0/    root)   Gid: (    0/    root)
Access: 2018-10-30 18:33:31.138550932 +0000
Modify: 2018-10-30 18:33:31.138550932 +0000
Change: 2018-10-30 18:33:31.138550932 +0000
 Birth: -
```

```
# cd #Acceder ao directorio casa do usuario (/home/user)
```

```
# umount /mnt/part1 #Desmontar (deixar de fazer uso) a partição primária /dev/sda1 que estava montada em /mnt/part1
```

```
# debugfs /dev/sda1 #Ejecutar o comando debugfs sobre a partição primaria /dev/sda1
```

```
debugfs 1.43.4 (31-Jan-2017)
debugfs: stat <16> #Ver información sobre o inodo 16
Inode: 16  Type: regular      Mode: 0644  Flags: 0x80000
Generation: 1746833275  Version: 0x00000000:00000001
User: 0  Group: 0  Project: 0  Size: 53
File ACL: 0  Directory ACL: 0
Links: 1  Blockcount: 8
Fragment: Address: 0  Number: 0  Size: 0
ctime: 0x5bd8a3fb:21087a50 -- Tue Oct 30 18:33:31 2018
atime: 0x5bd8a3fb:21087a50 -- Tue Oct 30 18:33:31 2018
mtime: 0x5bd8a3fb:21087a50 -- Tue Oct 30 18:33:31 2018
crtime: 0x5bd8a3fb:21087a50 -- Tue Oct 30 18:33:31 2018
Size of extra inode fields: 32
Inode checksum: 0x1c9445dc
EXTENTS:
(0):33281
(END)
```

Premer a tecla **q** para voltar á consola debugfs

```
debugfs: blocks <16> #Ver os bloques que apontam ao contido do ficheiro que corresponde co inodo 16
33281
```

```
debugfs: cat <16> #Ver o contido do ficheiro que corresponde co inodo 16
```

Cada día sabemos máis e entendemos menos. Einstein

```
debugfs: ncheck 16 #Identificar a relación entre o inodo 16 e o nome do arquivo ao que pertence.
```

```
Inode    Pathname
16       //proverbio1.txt
```

```
debugfs: icheck 33281 #Identificar a relación entre o bloque 33281 e o inodo ao que pertence.
```

Block	Inode number
1	1
2	2
3	3
4	4
5	5
6	6
7	7
8	8
9	9
10	10
11	11
12	12
13	13
14	14
15	15
16	16
17	17
18	18
19	19
20	20
21	21
22	22
23	23
24	24
25	25
26	26
27	27
28	28
29	29
30	30
31	31
32	32
33	33
34	34
35	35
36	36
37	37
38	38
39	39
40	40
41	41
42	42
43	43
44	44
45	45
46	46
47	47
48	48
49	49
50	50
51	51
52	52
53	53
54	54
55	55
56	56
57	57
58	58
59	59
60	60
61	61
62	62
63	63
64	64
65	65
66	66
67	67
68	68
69	69
70	70
71	71
72	72
73	73
74	74
75	75
76	76
77	77
78	78
79	79
80	80
81	81
82	82
83	83
84	84
85	85
86	86
87	87
88	88
89	89
90	90
91	91
92	92
93	93
94	94
95	95
96	96
97	97
98	98
99	99

33281	16
-------	----

```
debugfs: q Premer de novo a tecla q para sair da consola debugfs
```

```
# mount /dev/sda1 /mnt/part1 #Montar (facer uso) a partición primaria /dev/sda1 en /mnt/part1
```

```
# cd /mnt/part1 #Acceder ao directorio /mnt/part1
```

```
# for i in $(seq 1 100);do echo 'Cada día sabemos máis e entendemos menos. Einstein' >> proverbio2.txt;done #Crear un ficheiro /mnt/part1/proverbio2.txt que contén 100 frases.
```

```
# cat proverbio2.txt #Ver o contido do ficheiro /mnt/part1/proverbio2.txt
```

[illegible]



Premer a tecla **q** para voltar á consola debugfs

```
debugfs: cat <17> #Ver o contido do ficheiro que corresponde co inodo 17
```

45



```

Cada día sabemos máis e entendemos menos. Einstein
Cada día sabemos máis e entendemos menos. Einstein
Cada día sabemos máis e entendemos menos. Einstein
Cada día sabemos máis e entendemos menos. Einstein
Cada día sabemos máis e entendemos menos. Einstein
Cada día sabemos máis e entendemos menos. Einstein
Cada día sabemos máis e entendemos menos. Einstein
Cada día sabemos máis e entendemos menos. Einstein
Cada día sabemos máis e entendemos menos. Einstein
Cada día sabemos máis e entendemos menos. Einstein
Cada día sabemos máis e entendemos menos. Einstein
Cada día sabemos máis e entendemos menos. Einstein
Cada día sabemos máis e entendemos menos. Einstein
Cada día sabemos máis e entendemos menos. Einstein
debugfs: ncheck 17 #Identificar a relación entre o inodo 17 e o nome do arquivo ao que pertence.
Inode Pathname
17 //proverbio2.txt
debugfs: icheck 33794 #Identificar a relación entre o bloque 33281 e o inodo ao que pertence.
Block Inode number
33794 17
debugfs: icheck 33795 #Identificar a relación entre o bloque 33281 e o inodo ao que pertence.
Block Inode number
33795 17
debugfs: block_dump 33794 #Ver o contido do bloque 33794. A información amósase en código hexadecimal e ASCII

0000 4361 6461 2064 c3ad 6120 7361 6265 6d6f Cada d..a sabemo
0020 7320 6dc3 a169 7320 6520 656e 7465 6e64 s m..is e entend
0040 656d 6f73 206d 656e 6f73 2e20 4569 6e73 emos menos. Eins
0060 7465 696e 0a43 6164 6120 64c3 ad61 2073 tein.Cada d..a s
0100 6162 656d 6f73 206d c3a1 6973 2065 2065 abemos m..is e e
0120 6e74 656e 6465 6d6f 7320 6d65 6e6f 732e ntendemos menos.
0140 2045 696e 7374 6569 6e0a 4361 6461 2064 Einstein.Cada d
0160 c3ad 6120 7361 6265 6d6f 7320 6dc3 a169 ..a sabemos m..i
0200 7320 6520 656e 7465 6e64 656d 6f73 206d s e entendemos m
0220 656e 6f73 2e20 4569 6e73 7465 696e 0a43 enos. Einstein.C
0240 6164 6120 64c3 ad61 2073 6162 656d 6f73 ada d..a sabemos
0260 206d c3a1 6973 2065 2065 6e74 656e 6465 m..is e entende
0300 6d6f 7320 6d65 6e6f 732e 2045 696e 7374 mos menos. Einst
0320 6569 6e0a 4361 6461 2064 c3ad 6120 7361 ein.Cada d..a sa
0340 6265 6d6f 7320 6dc3 a169 7320 6520 656e bemos m..is e en
0360 7465 6e64 656d 6f73 206d 656e 6f73 2e20 tendemos menos.
0400 4569 6e73 7465 696e 0a43 6164 6120 64c3 Einstein.Cada d.
0420 ad61 2073 6162 656d 6f73 206d c3a1 6973 .a sabemos m..is
0440 2065 2065 6e74 656e 6465 6d6f 7320 6d65 e entendemos me
0460 6e6f 732e 2045 696e 7374 6569 6e0a 4361 nos. Einstein.Ca
0500 6461 2064 c3ad 6120 7361 6265 6d6f 7320 da d..a sabemos
0520 6dc3 a169 7320 6520 656e 7465 6e64 656d m..is e entendem
0540 6f73 206d 656e 6f73 2e20 4569 6e73 7465 os menos. Einste
0560 696e 0a43 6164 6120 64c3 ad61 2073 6162 in.Cada d..a sab
0600 656d 6f73 206d c3a1 6973 2065 2065 6e74 emos m..is e ent
0620 656e 6465 6d6f 7320 6d65 6e6f 732e 2045 endemos menos. E
0640 696e 7374 6569 6e0a 4361 6461 2064 c3ad instein.Cada d..
0660 6120 7361 6265 6d6f 7320 6dc3 a169 7320 a sabemos m..is
0700 6520 656e 7465 6e64 656d 6f73 206d 656e e entendemos men
0720 6f73 2e20 4569 6e73 7465 696e 0a43 6164 os. Einstein.Cad
0740 6120 64c3 ad61 2073 6162 656d 6f73 206d a d..a sabemos m
0760 c3a1 6973 2065 2065 6e74 656e 6465 6d6f ..is e entendem
1000 7320 6d65 6e6f 732e 2045 696e 7374 6569 s menos. Einstei
1020 6e0a 4361 6461 2064 c3ad 6120 7361 6265 n.Cada d..a sabe
1040 6d6f 7320 6dc3 a169 7320 6520 656e 7465 mos m..is e ente
1060 6e64 656d 6f73 206d 656e 6f73 2e20 4569 ndemos menos. Ei
1100 6e73 7465 696e 0a43 6164 6120 64c3 ad61 nstein.Cada d..a
1120 2073 6162 656d 6f73 206d c3a1 6973 2065 sabemos m..is e
1140 2065 6e74 656e 6465 6d6f 7320 6d65 6e6f entendemos meno
1160 732e 2045 696e 7374 6569 6e0a 4361 6461 s. Einstein.Cada
1200 2064 c3ad 6120 7361 6265 6d6f 7320 6dc3 d..a sabemos m.
1220 a169 7320 6520 656e 7465 6e64 656d 6f73 .is e entendemos
1240 206d 656e 6f73 2e20 4569 6e73 7465 696e menos. Einstein
1260 0a43 6164 6120 64c3 ad61 2073 6162 656d .Cada d..a sabem
1300 6f73 206d c3a1 6973 2065 2065 6e74 656e os m..is e enten
1320 6465 6d6f 7320 6d65 6e6f 732e 2045 696e demos menos. Ein
1340 7374 6569 6e0a 4361 6461 2064 c3ad 6120 stein.Cada d..a
1360 7361 6265 6d6f 7320 6dc3 a169 7320 6520 sabemos m..is e
1400 656e 7465 6e64 656d 6f73 206d 656e 6f73 entendemos menos
1420 2e20 4569 6e73 7465 696e 0a43 6164 6120 . Einstein.Cada
1440 64c3 ad61 2073 6162 656d 6f73 206d c3a1 d..a sabemos m..
1460 6973 2065 2065 6e74 656e 6465 6d6f 7320 is e entendemos
1500 6d65 6e6f 732e 2045 696e 7374 6569 6e0a menos. Einstein.
1520 4361 6461 2064 c3ad 6120 7361 6265 6d6f Cada d..a sabemo
1540 7320 6dc3 a169 7320 6520 656e 7465 6e64 s m..is e entend
1560 656d 6f73 206d 656e 6f73 2e20 4569 6e73 emos menos. Eins
1600 7465 696e 0a43 6164 6120 64c3 ad61 2073 tein.Cada d..a s
1620 6162 656d 6f73 206d c3a1 6973 2065 2065 abemos m..is e e
1640 6e74 656e 6465 6d6f 7320 6d65 6e6f 732e ntendemos menos.
1660 2045 696e 7374 6569 6e0a 4361 6461 2064 Einstein.Cada d
1700 c3ad 6120 7361 6265 6d6f 7320 6dc3 a169 ..a sabemos m..i
1720 7320 6520 656e 7465 6e64 656d 6f73 206d s e entendemos m
1740 656e 6f73 2e20 4569 6e73 7465 696e 0a43 enos. Einstein.C
1760 6164 6120 64c3 ad61 2073 6162 656d 6f73 ada d..a sabemos
2000 206d c3a1 6973 2065 2065 6e74 656e 6465 m..is e entende
2020 6d6f 7320 6d65 6e6f 732e 2045 696e 7374 mos menos. Einst
2040 6569 6e0a 4361 6461 2064 c3ad 6120 7361 ein.Cada d..a sa
2060 6265 6d6f 7320 6dc3 a169 7320 6520 656e bemos m..is e en
2100 7465 6e64 656d 6f73 206d 656e 6f73 2e20 tendemos menos.
2120 4569 6e73 7465 696e 0a43 6164 6120 64c3 Einstein.Cada d.
2140 ad61 2073 6162 656d 6f73 206d c3a1 6973 .a sabemos m..is
2160 2065 2065 6e74 656e 6465 6d6f 7320 6d65 e entendemos me
2200 6e6f 732e 2045 696e 7374 6569 6e0a 4361 nos. Einstein.Ca
2220 6461 2064 c3ad 6120 7361 6265 6d6f 7320 da d..a sabemos
2240 6dc3 a169 7320 6520 656e 7465 6e64 656d m..is e entendem
2260 6f73 206d 656e 6f73 2e20 4569 6e73 7465 os menos. Einste

```

2300 696e 0a43 6164 6120 64c3 ad61 2073 6162 in.Cada d..a sab  
2320 656d 6f73 206d c3a1 6973 2065 2065 6e74 emos m..is e ent  
2340 656e 6465 6d6f 7320 6d65 6e6f 732e 2045 endemos menos. E  
2360 696e 7374 6569 6e0a 4361 6461 2064 c3ad instein.Cada d..  
2400 6120 7361 6265 6d6f 7320 6dc3 a169 7320 a sabemos m..is  
2420 6520 656e 7465 6e64 656d 6f73 206d 656e e entendemos men  
2440 6f73 2e20 4569 6e73 7465 696e 0a43 6164 os. Einstein.Cad  
2460 6120 64c3 ad61 2073 6162 656d 6f73 206d a d..a sabemos m  
2500 c3a1 6973 2065 2065 6e74 656e 6465 6d6f ..is e entendemo  
2520 7320 6d65 6e6f 732e 2045 696e 7374 6569 s menos. Einstei  
2540 6e0a 4361 6461 2064 c3ad 6120 7361 6265 n.Cada d..a sabe  
2560 6d6f 7320 6dc3 a169 7320 6520 656e 7465 mos m..is e ente  
2600 6e64 656d 6f73 206d 656e 6f73 2e20 4569 ndemos menos. Ei  
2620 6e73 7465 696e 0a43 6164 6120 64c3 ad61 nstein.Cada d..a  
2640 2073 6162 656d 6f73 206d c3a1 6973 2065 sabemos m..is e  
2660 2065 6e74 656e 6465 6d6f 7320 6d65 6e6f entendemos meno  
2700 732e 2045 696e 7374 6569 6e0a 4361 6461 s. Einstein.Cada  
2720 2064 c3ad 6120 7361 6265 6d6f 7320 6dc3 d..a sabemos m.  
2740 a169 7320 6520 656e 7465 6e64 656d 6f73 ..is e entendemos  
2760 206d 656e 6f73 2e20 4569 6e73 7465 696e menos. Einstein  
3000 0a43 6164 6120 64c3 ad61 2073 6162 656d .Cada d..a sabem  
3020 6f73 206d c3a1 6973 2065 2065 6e74 656e os m..is e enten  
3040 6465 6d6f 7320 6d65 6e6f 732e 2045 696e demos menos. Ein  
3060 7374 6569 6e0a 4361 6461 2064 c3ad 6120 stein.Cada d..a  
3100 7361 6265 6d6f 7320 6dc3 a169 7320 6520 sabemos m..is e  
3120 656e 7465 6e64 656d 6f73 206d 656e 6f73 entendemos menos  
3140 2e20 4569 6e73 7465 696e 0a43 6164 6120 . Einstein.Cada  
3160 64c3 ad61 2073 6162 656d 6f73 206d c3a1 d..a sabemos m..  
3200 6973 2065 2065 6e74 656e 6465 6d6f 7320 is e entendemos  
3220 6d65 6e6f 732e 2045 696e 7374 6569 6e0a menos. Einstein.  
3240 4361 6461 2064 c3ad 6120 7361 6265 6d6f Cada d..a sabemo  
3260 7320 6dc3 a169 7320 6520 656e 7465 6e64 s m..is e entend  
3300 656d 6f73 206d 656e 6f73 2e20 4569 6e73 emos menos. Eins  
3320 7465 696e 0a43 6164 6120 64c3 ad61 2073 tein.Cada d..a s  
3340 6162 656d 6f73 206d c3a1 6973 2065 2065 abemos m..is e e  
3360 6e74 656e 6465 6d6f 7320 6d65 6e6f 732e ntendemos menos.  
3400 2045 696e 7374 6569 6e0a 4361 6461 2064 Einstein.Cada d  
3420 c3ad 6120 7361 6265 6d6f 7320 6dc3 a169 ..a sabemos m..i  
3440 7320 6520 656e 7465 6e64 656d 6f73 206d s e entendemos m  
3460 656e 6f73 2e20 4569 6e73 7465 696e 0a43 enos. Einstein.C  
3500 6164 6120 64c3 ad61 2073 6162 656d 6f73 ada d..a sabemos  
3520 206d c3a1 6973 2065 2065 6e74 656e 6465 m..is e entende  
3540 6d6f 7320 6d65 6e6f 732e 2045 696e 7374 mos menos. Einst  
3560 6569 6e0a 4361 6461 2064 c3ad 6120 7361 ein.Cada d..a sa  
3600 6265 6d6f 7320 6dc3 a169 7320 6520 656e bemos m..is e en  
3620 7465 6e64 656d 6f73 206d 656e 6f73 2e20 tendemos menos.  
3640 4569 6e73 7465 696e 0a43 6164 6120 64c3 Einstein.Cada d.  
3660 ad61 2073 6162 656d 6f73 206d c3a1 6973 .a sabemos m..is  
3700 2065 2065 6e74 656e 6465 6d6f 7320 6d65 e entendemos me  
3720 6e6f 732e 2045 696e 7374 6569 6e0a 4361 nos. Einstein.Ca  
3740 6461 2064 c3ad 6120 7361 6265 6d6f 7320 da d..a sabemos  
3760 6dc3 a169 7320 6520 656e 7465 6e64 656d m..is e entendem  
4000 6f73 206d 656e 6f73 2e20 4569 6e73 7465 os menos. Einste  
4020 696e 0a43 6164 6120 64c3 ad61 2073 6162 in.Cada d..a sab  
4040 656d 6f73 206d c3a1 6973 2065 2065 6e74 emos m..is e ent  
4060 656e 6465 6d6f 7320 6d65 6e6f 732e 2045 endemos menos. E  
4100 696e 7374 6569 6e0a 4361 6461 2064 c3ad instein.Cada d..  
4120 6120 7361 6265 6d6f 7320 6dc3 a169 7320 a sabemos m..is  
4140 6520 656e 7465 6e64 656d 6f73 206d 656e e entendemos men  
4160 6f73 2e20 4569 6e73 7465 696e 0a43 6164 os. Einstein.Cad  
4200 6120 64c3 ad61 2073 6162 656d 6f73 206d a d..a sabemos m  
4220 c3a1 6973 2065 2065 6e74 656e 6465 6d6f ..is e entendemo  
4240 7320 6d65 6e6f 732e 2045 696e 7374 6569 s menos. Einstei  
4260 6e0a 4361 6461 2064 c3ad 6120 7361 6265 n.Cada d..a sabe  
4300 6d6f 7320 6dc3 a169 7320 6520 656e 7465 mos m..is e ente  
4320 6e64 656d 6f73 206d 656e 6f73 2e20 4569 ndemos menos. Ei  
4340 6e73 7465 696e 0a43 6164 6120 64c3 ad61 nstein.Cada d..a  
4360 2073 6162 656d 6f73 206d c3a1 6973 2065 sabemos m..is e  
4400 2065 6e74 656e 6465 6d6f 7320 6d65 6e6f entendemos meno  
4420 732e 2045 696e 7374 6569 6e0a 4361 6461 s. Einstein.Cada  
4440 2064 c3ad 6120 7361 6265 6d6f 7320 6dc3 d..a sabemos m.  
4460 a169 7320 6520 656e 7465 6e64 656d 6f73 ..is e entendemos  
4500 206d 656e 6f73 2e20 4569 6e73 7465 696e menos. Einstein  
4520 0a43 6164 6120 64c3 ad61 2073 6162 656d .Cada d..a sabem  
4540 6f73 206d c3a1 6973 2065 2065 6e74 656e os m..is e enten  
4560 6465 6d6f 7320 6d65 6e6f 732e 2045 696e demos menos. Ein  
4600 7374 6569 6e0a 4361 6461 2064 c3ad 6120 stein.Cada d..a  
4620 7361 6265 6d6f 7320 6dc3 a169 7320 6520 sabemos m..is e  
4640 656e 7465 6e64 656d 6f73 206d 656e 6f73 entendemos menos  
4660 2e20 4569 6e73 7465 696e 0a43 6164 6120 . Einstein.Cada  
4700 64c3 ad61 2073 6162 656d 6f73 206d c3a1 d..a sabemos m..  
4720 6973 2065 2065 6e74 656e 6465 6d6f 7320 is e entendemos  
4740 6d65 6e6f 732e 2045 696e 7374 6569 6e0a menos. Einstein.  
4760 4361 6461 2064 c3ad 6120 7361 6265 6d6f Cada d..a sabemo  
5000 7320 6dc3 a169 7320 6520 656e 7465 6e64 s m..is e entend  
5020 656d 6f73 206d 656e 6f73 2e20 4569 6e73 emos menos. Eins  
5040 7465 696e 0a43 6164 6120 64c3 ad61 2073 tein.Cada d..a s  
5060 6162 656d 6f73 206d c3a1 6973 2065 2065 abemos m..is e e  
5100 6e74 656e 6465 6d6f 7320 6d65 6e6f 732e ntendemos menos.  
5120 2045 696e 7374 6569 6e0a 4361 6461 2064 Einstein.Cada d  
5140 c3ad 6120 7361 6265 6d6f 7320 6dc3 a169 ..a sabemos m..i  
5160 7320 6520 656e 7465 6e64 656d 6f73 206d s e entendemos m  
5200 656e 6f73 2e20 4569 6e73 7465 696e 0a43 enos. Einstein.C  
5220 6164 6120 64c3 ad61 2073 6162 656d 6f73 ada d..a sabemos  
5240 206d c3a1 6973 2065 2065 6e74 656e 6465 m..is e entende  
5260 6d6f 7320 6d65 6e6f 732e 2045 696e 7374 mos menos. Einst  
5300 6569 6e0a 4361 6461 2064 c3ad 6120 7361 ein.Cada d..a sa  
5320 6265 6d6f 7320 6dc3 a169 7320 6520 656e bemos m..is e en  
5340 7465 6e64 656d 6f73 206d 656e 6f73 2e20 tendemos menos.  
5360 4569 6e73 7465 696e 0a43 6164 6120 64c3 Einstein.Cada d.  
5400 ad61 2073 6162 656d 6f73 206d c3a1 6973 .a sabemos m..is

5420 2065 2065 6e74 656e 6465 6d6f 7320 6d65 e entendemos me  
5440 6e6f 732e 2045 696e 7374 6569 6e0a 4361 nos. Einstein.Ca  
5460 6461 2064 c3ad 6120 7361 6265 6d6f 7320 da d..a sabemos  
5500 6dc3 a169 7320 6520 656e 7465 6e64 656d m..is e entendem  
5520 c3a1 206d 656e 6f73 2e20 4569 6e73 7465 os menos. Einste  
5540 696e 0a43 6164 6120 64c3 ad61 2073 6162 in.Cada d..a sab  
5560 656d 6f73 206d c3a1 6973 2065 2065 6e74 emos m..is e ent  
5600 656e 6465 6d6f 7320 6d65 6e6f 732e 2045 endemos menos. E  
5620 696e 7374 6569 6e0a 4361 6461 2064 c3ad instein.Cada d..  
5640 6120 7361 6265 6d6f 7320 6dc3 a169 7320 a sabemos m..is  
5660 6520 656e 7465 6e64 656d 6f73 206d 656e e entendemos men  
5700 6f73 2e20 4569 6e73 7465 696e 0a43 6164 os. Einstein.Cad  
5720 6120 64c3 ad61 2073 6162 656d 6f73 206d a d..a sabemos m  
5740 c3a1 6973 2065 2065 6e74 656e 6465 6d6f ..is e entendemo  
5760 7320 6d65 6e6f 732e 2045 696e 7374 6569 s menos. Einstei  
6000 6e0a 4361 6461 2064 c3ad 6120 7361 6265 n.Cada d..a sabe  
6020 6d6f 7320 6dc3 a169 7320 6520 656e 7465 mos m..is e ente  
6040 6e64 656d 6f73 206d 656e 6f73 2e20 4569 ndemos menos. Ei  
6060 6e73 7465 696e 0a43 6164 6120 64c3 ad61 nstein.Cada d..a  
6100 2073 6162 656d 6f73 206d c3a1 6973 2065 sabemos m..is e  
6120 2065 6e74 656e 6465 6d6f 7320 6d65 6e6f entendemos meno  
6140 732e 2045 696e 7374 6569 6e0a 4361 6461 s. Einstein.Cada  
6160 2064 c3ad 6120 7361 6265 6d6f 7320 6dc3 d..a sabemos m.  
6200 a169 7320 6520 656e 7465 6e64 656d 6f73 .is e entendemos  
6220 206d 656e 6f73 2e20 4569 6e73 7465 696e menos. Einstein  
6240 0a43 6164 6120 64c3 ad61 2073 6162 656d .Cada d..a sabem  
6260 6f73 206d c3a1 6973 2065 2065 6e74 656e os m..is e enten  
6300 6465 6d6f 7320 6d65 6e6f 732e 2045 696e demos menos. Ein  
6320 7374 6569 6e0a 4361 6461 2064 c3ad 6120 stein.Cada d..a  
6340 7361 6265 6d6f 7320 6dc3 a169 7320 6520 sabemos m..is e  
6360 656e 7465 6e64 656d 6f73 206d 656e 6f73 entendemos menos  
6400 2e20 4569 6e73 7465 696e 0a43 6164 6120 . Einstein.Cada  
6420 64c3 ad61 2073 6162 656d 6f73 206d c3a1 d..a sabemos m..  
6440 6973 2065 2065 6e74 656e 6465 6d6f 7320 is e entendemos  
6460 6d65 6e6f 732e 2045 696e 7374 6569 6e0a menos. Einstein.  
6500 4361 6461 2064 c3ad 6120 7361 6265 6d6f Cada d..a sabemo  
6520 7320 6dc3 a169 7320 6520 656e 7465 6e64 s m..is e entend  
6540 656d 6f73 206d 656e 6f73 2e20 4569 6e73 emos menos. Eins  
6560 7465 696e 0a43 6164 6120 64c3 ad61 2073 tein.Cada d..a s  
6600 6162 656d 6f73 206d c3a1 6973 2065 2065 abemos m..is e e  
6620 6e74 656e 6465 6d6f 7320 6d65 6e6f 732e ntendemos menos.  
6640 2045 696e 7374 6569 6e0a 4361 6461 2064 Einstein.Cada d  
6660 c3ad 6120 7361 6265 6d6f 7320 6dc3 a169 ..a sabemos m..i  
6700 7320 6520 656e 7465 6e64 656d 6f73 206d s e entendemos m  
6720 656e 6f73 2e20 4569 6e73 7465 696e 0a43 enos. Einstein.C  
6740 6164 6120 64c3 ad61 2073 6162 656d 6f73 ada d..a sabemos  
6760 206d c3a1 6973 2065 2065 6e74 656e 6465 m..is e entende  
7000 6d6f 7320 6d65 6e6f 732e 2045 696e 7374 mos menos. Einst  
7020 6569 6e0a 4361 6461 2064 c3ad 6120 7361 ein.Cada d..a sa  
7040 6265 6d6f 7320 6dc3 a169 7320 6520 656e bemos m..is e en  
7060 7465 6e64 656d 6f73 206d 656e 6f73 2e20 tendemos menos.  
7100 4569 6e73 7465 696e 0a43 6164 6120 64c3 Einstein.Cada d.  
7120 ad61 2073 6162 656d 6f73 206d c3a1 6973 .a sabemos m..is  
7140 2065 2065 6e74 656e 6465 6d6f 7320 6d65 e entendemos me  
7160 6e6f 732e 2045 696e 7374 6569 6e0a 4361 nos. Einstein.Ca  
7200 6461 2064 c3ad 6120 7361 6265 6d6f 7320 da d..a sabemos  
7220 6dc3 a169 7320 6520 656e 7465 6e64 656d m..is e entendem  
7240 6f73 206d 656e 6f73 2e20 4569 6e73 7465 os menos. Einste  
7260 696e 0a43 6164 6120 64c3 ad61 2073 6162 in.Cada d..a sab  
7300 656d 6f73 206d c3a1 6973 2065 2065 6e74 emos m..is e ent  
7320 656e 6465 6d6f 7320 6d65 6e6f 732e 2045 endemos menos. E  
7340 696e 7374 6569 6e0a 4361 6461 2064 c3ad instein.Cada d..  
7360 6120 7361 6265 6d6f 7320 6dc3 a169 7320 a sabemos m..is  
7400 6520 656e 7465 6e64 656d 6f73 206d 656e e entendemos men  
7420 6f73 2e20 4569 6e73 7465 696e 0a43 6164 os. Einstein.Cad  
7440 6120 64c3 ad61 2073 6162 656d 6f73 206d a d..a sabemos m  
7460 c3a1 6973 2065 2065 6e74 656e 6465 6d6f ..is e entendemo  
7500 7320 6d65 6e6f 732e 2045 696e 7374 6569 s menos. Einstei  
7520 6e0a 4361 6461 2064 c3ad 6120 7361 6265 n.Cada d..a sabe  
7540 6d6f 7320 6dc3 a169 7320 6520 656e 7465 mos m..is e ente  
7560 6e64 656d 6f73 206d 656e 6f73 2e20 4569 ndemos menos. Ei  
7600 6e73 7465 696e 0a43 6164 6120 64c3 ad61 nstein.Cada d..a  
7620 2073 6162 656d 6f73 206d c3a1 6973 2065 sabemos m..is e  
7640 2065 6e74 656e 6465 6d6f 7320 6d65 6e6f entendemos meno  
7660 732e 2045 696e 7374 6569 6e0a 4361 6461 s. Einstein.Cada  
7700 2064 c3ad 6120 7361 6265 6d6f 7320 6dc3 d..a sabemos m.  
7720 a169 7320 6520 656e 7465 6e64 656d 6f73 .is e entendemos  
7740 206d 656e 6f73 2e20 4569 6e73 7465 696e menos. Einstein  
7760 0a43 6164 6120 64c3 ad61 2073 6162 656d .Cada d..a sabem

debugfs: q **Premar de novo a tecla q para sair da consola debugfs**

Ricardo Feijoo Costa



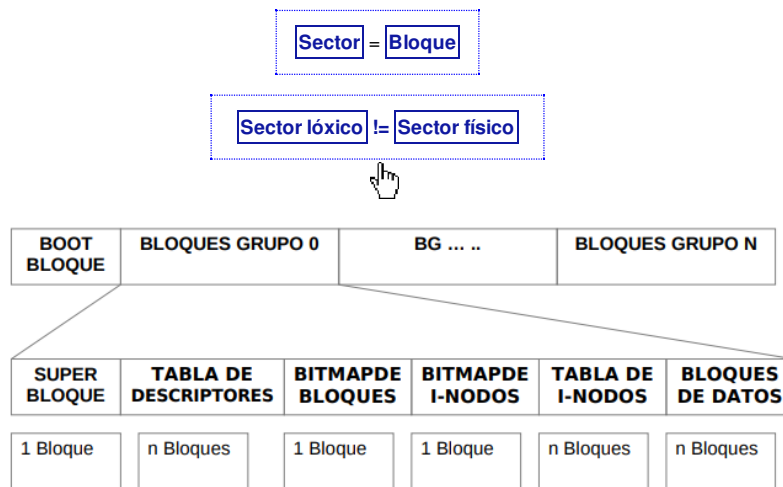
This work is licensed under a **Creative Commons Attribution-ShareAlike 4.0 International License**



# Sistemas de arquivos: Superbloque

## Información sobre o sistema de ficheiros

NOTA: Para verificar o que acontece na práctica crear unha máquina virtual en VirtualBox que arranque cunha ISO Live Debian 32bits, escritorio XFCE, 512MB de RAM e disco duro dinámico de 8GB. Imos supor que esta máquina virtual posúe o nome **Debian32-Recovery** e o disco duro posúe o nome **Debian32-Recovery.vdi**. Verificar que a primeira opción de arranque sexa o CD Virtual.



Diseño de Sistemas Operativos. Universidad de Las Palmas de Gran Canaria

O **Superbloque** é un bloque que contén a información máis relevante e describe o sistema de ficheiros.

Cada grupo de bloques contén unha copia do superbloque.

O superbloque contén 3 tipos de parámetros:

1. **Parámetros non modificables:** Creados cando se xera o sistema de ficheiros.
  - o Tamaño de bloque
  - o Número de bloques dispoñibles
  - o Número de inodos dispoñibles
2. **Parámetros modificables** durante o uso do sistema de ficheiros:
  - o Bloques reservados ao superusuario
  - o UID (identificador usuario) do superusuario por defecto
  - o GID (identificador de grupo) do superusuario por defecto
3. **Parámetros de estado e uso** que varían durante o uso do sistema de ficheiros:
  - o Inodos e bloques libres
  - o Estado do sistemas de ficheiros (limpo/non limpo)
  - o Número de montaxes
  - o Máximo número de montaxes entre comprobacións
  - o Máximo tempo entre comprobacións
  - o Comportamento fronte a erros
  - o Directorio da última montaxe

### Comandos de interese sobre sistemas de ficheiros ext2/ext3/ext4

- **dumpe2fs** #O comando **dumpe2fs** permite listar información sobre sistemas de ficheiros ext2/ext3/ext4. Aмосa información sobre o superbloque e grupos de bloques.
- **mke2fs** #O comando **mke2fs** permite crear sistemas de ficheiros ext2/ext3/ext4. Coa opción **-n** non xera sistemas de ficheiros pero permite amosar a localización dos bloques de backup do superbloque.
- **debugfs** #O comando **debugfs** permite depurar sistemas de ficheiros ext2/ext3/ext4

### Amosar información sobre os superbloques e grupos de bloques

```
# dumpe2fs /dev/sda1 #Amosar información sobre o superbloque e grupos de bloques da partición /dev/sda1

# mke2fs -n /dev/sda1 #Amosar a lista dos superbloques, é dicir cales son os bloques que gardan a información do
superbloque en /dev/sda1

# dumpe2fs /dev/sda2 #Amosar información sobre o superbloque e grupos de bloques da partición /dev/sda2

# mke2fs -n /dev/sda2 #Amosar a lista dos superbloques, é dicir cales son os bloques que gardan a información do
superbloque en /dev/sda2
```

## Práctica

1. Arrancar a máquina virtual creada en modo Inicio normal

**VBoxManage startvm Debian32-Recovery**

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

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

```
$ sudo su - #Acceder á consola de root(administrador) a través dos permisos configurados co comando sudo (/etc/sudoers, visudo)
```

### Crear e formatear particións

```
# parted --script /dev/sda mklabel msdos #Crear a etiqueta de disco (táboa de particións) ao dispositivo /dev/sda sen ter que acceder ao prompt de parted
```

```
# parted --script /dev/sda mkpart primary 0 50% -a cylinder #Crear unha partición primaria no disco /dev/sda cos primeiros 5GB, alineando a cilindros, sen ter que acceder ao prompt de parted
```

```
# parted --script /dev/sda mkpart primary 50% 70% -a cylinder #Crear unha partición primaria no disco /dev/sda de 2GB a continuación da partición de 5GB, alineando a cilindros, sen ter que acceder ao prompt de parted
```

```
# parted --script /dev/sda print #Amosa a táboa de particións do disco /dev/sda
```

Model: ATA VBOX HARDDISK (scsi)

Disk /dev/sda: 8590MB

Sector size (logical/physical): 512B/512B

Partition Table: msdos

Disk Flags:

Number	Start	End	Size	Type	File system	Flags
1	32.3kB	4294MB	4294MB	primary	ext4	
2	4294MB	6013MB	1719MB	primary	ext3	

```
# mkfs.ext4 -F -L 'PARTICION1' /dev/sda1 #Formatear en ext4 coa etiqueta PARTICION1 a partición primaria /dev/sda1
```

mke2fs 1.43.4 (31-Jan-2017)

/dev/sda1 contains a ext4 file system labelled 'PARTICION1'

last mounted on /mnt/recuperacion on Mon Oct 29 17:58:50 2018

Creating filesystem with 1048233 4k blocks and 262144 inodes

Filesystem UUID: 459fb916-7189-4b3f-83de-fd81b56973f8

Superblock backups stored on blocks:

32768, 98304, 163840, 229376, 294912, 819200, 884736

Allocating group tables: done

Writing inode tables: done

Creating journal (16384 blocks): done

Writing superblocks and filesystem accounting information: done

```
# mkfs.ext3 -F -L 'PARTICION2' /dev/sda2 #Formatear en ext3 coa etiqueta PARTICION2 a partición primaria /dev/sda2
```

mke2fs 1.43.4 (31-Jan-2017)

/dev/sda2 contains a ext3 file system labelled 'PARTICION2'

created on Mon Oct 29 17:58:29 2018

Creating filesystem with 419698 4k blocks and 105040 inodes

Filesystem UUID: d5c70817-fa3c-4734-8825-fa78463558cd

Superblock backups stored on blocks:

32768, 98304, 163840, 229376, 294912

Allocating group tables: done

Writing inode tables: done

Creating journal (8192 blocks): done

Writing superblocks and filesystem accounting information: done

### Amosar información sobre os superbloques e grupos de bloques

```
# dumpe2fs /dev/sda1 #Amosar información sobre o superbloque e grupos de bloques da partición /dev/sda1
```

dumpe2fs 1.43.4 (31-Jan-2017)

Filesystem volume name: PARTICION1

Last mounted on: /mnt/recuperacion

Filesystem UUID: ad1260c3-bc25-4a42-81a2-0ef0090b006c

Filesystem magic number: 0xEF53

Filesystem revision #: 1 (dynamic)

Filesystem features: has\_journal ext\_attr resize\_inode dir\_index filetype extent 64bit flex\_bg sparse\_super large\_file huge\_file dir\_nlink extra\_isize metadata\_csum

Filesystem flags: signed\_directory\_hash

Default mount options: user\_xattr acl

Filesystem state: clean

Errors behavior: Continue

Filesystem OS type: Linux

Inode count: 262144

Block count: 1048233

Reserved block count: 52411

Free blocks: 1011288

Free inodes: 262130

First block: 0

Block size: 4096

Fragment size: 4096

Group descriptor size: 64

Reserved GDT blocks: 511

Blocks per group: 32768

Fragments per group: 32768

Inodes per group: 8192

Inode blocks per group: 512

Flex block group size: 16

Filesystem created: Thu Nov 1 21:22:37 2018

Last mount time: Sun Nov 4 18:47:07 2018  
Last write time: Sun Nov 4 18:47:23 2018  
Mount count: 7  
Maximum mount count: -1  
Last checked: Thu Nov 1 21:22:37 2018  
Check interval: 0 ()  
Lifetime writes: 131 MB  
Reserved blocks uid: 0 (user root)  
Reserved blocks gid: 0 (group root)  
First inode: 11  
Inode size: 256  
Required extra isize: 32  
Desired extra isize: 32  
Journal inode: 8  
Default directory hash: half\_md4  
Directory Hash Seed: 75f31c99-c547-4769-b9a7-4550756391a7  
Journal backup: inode blocks  
Checksum type: crc32c  
Checksum: 0xc8fc6d4a  
Journal features: journal\_64bit journal\_checksum\_v3  
Journal size: 64M  
Journal length: 16384  
Journal sequence: 0x00000020  
Journal start: 0  
Journal checksum type: crc32c  
Journal checksum: 0x511ffe0b

Group 0: (Blocks 0-32767) csum 0x4ddc [ITABLE\_ZEROED]

Primary superblock at 0, Group descriptors at 1-1

Reserved GDT blocks at 2-512

Block bitmap at 513 (+513), csum 0x53ee830b

Inode bitmap at 529 (+529), csum 0x7fe90404

Inode table at 545-1056 (+545)

24025 free blocks, 8181 free inodes, 2 directories, 8181 unused inodes

Free blocks: 8743-32767

Free inodes: 12-8192

Group 1: (Blocks 32768-65535) csum 0xa9c0 [INODE\_UNINIT, BLOCK\_UNINIT, ITABLE\_ZEROED]

Backup superblock at 32768, Group descriptors at 32769-32769

Reserved GDT blocks at 32770-33280

Block bitmap at 514 (bg #0 + 514), csum 0x00000000

Inode bitmap at 530 (bg #0 + 530), csum 0x00000000

Inode table at 1057-1568 (bg #0 + 1057)

32255 free blocks, 8192 free inodes, 0 directories, 8192 unused inodes

Free blocks: 33281-65535

Free inodes: 8193-16384

Group 2: (Blocks 65536-98303) csum 0x8d30 [INODE\_UNINIT, BLOCK\_UNINIT, ITABLE\_ZEROED]

Block bitmap at 515 (bg #0 + 515), csum 0x00000000

Inode bitmap at 531 (bg #0 + 531), csum 0x00000000

Inode table at 1569-2080 (bg #0 + 1569)

32768 free blocks, 8192 free inodes, 0 directories, 8192 unused inodes

Free blocks: 65536-98303

Free inodes: 16385-24576

Group 3: (Blocks 98304-131071) csum 0xe0c5 [INODE\_UNINIT, BLOCK\_UNINIT, ITABLE\_ZEROED]

Backup superblock at 98304, Group descriptors at 98305-98305

Reserved GDT blocks at 98306-98816

Block bitmap at 516 (bg #0 + 516), csum 0x00000000

Inode bitmap at 532 (bg #0 + 532), csum 0x00000000

Inode table at 2081-2592 (bg #0 + 2081)

32255 free blocks, 8192 free inodes, 0 directories, 8192 unused inodes

Free blocks: 98817-131071

Free inodes: 24577-32768

Group 4: (Blocks 131072-163839) csum 0x52d9 [INODE\_UNINIT, BLOCK\_UNINIT, ITABLE\_ZEROED]

Block bitmap at 517 (bg #0 + 517), csum 0x00000000

Inode bitmap at 533 (bg #0 + 533), csum 0x00000000

Inode table at 2593-3104 (bg #0 + 2593)

32768 free blocks, 8192 free inodes, 0 directories, 8192 unused inodes

Free blocks: 131072-163839

Free inodes: 32769-40960

Group 5: (Blocks 163840-196607) csum 0x3a8e [INODE\_UNINIT, BLOCK\_UNINIT, ITABLE\_ZEROED]

Backup superblock at 163840, Group descriptors at 163841-163841

Reserved GDT blocks at 163842-164352

Block bitmap at 518 (bg #0 + 518), csum 0x00000000

Inode bitmap at 534 (bg #0 + 534), csum 0x00000000

Inode table at 3105-3616 (bg #0 + 3105)

32255 free blocks, 8192 free inodes, 0 directories, 8192 unused inodes

Free blocks: 164353-196607

Free inodes: 40961-49152

Group 6: (Blocks 196608-229375) csum 0x187e [INODE\_UNINIT, BLOCK\_UNINIT, ITABLE\_ZEROED]

Block bitmap at 519 (bg #0 + 519), csum 0x00000000

Inode bitmap at 535 (bg #0 + 535), csum 0x00000000

Inode table at 3617-4128 (bg #0 + 3617)

32768 free blocks, 8192 free inodes, 0 directories, 8192 unused inodes

Free blocks: 196608-229375

Free inodes: 49153-57344

Group 7: (Blocks 229376-262143) csum 0x083e [INODE\_UNINIT, BLOCK\_UNINIT, ITABLE\_ZEROED]

Backup superblock at 229376, Group descriptors at 229377-229377

Reserved GDT blocks at 229378-229888

Block bitmap at 520 (bg #0 + 520), csum 0x00000000

Inode bitmap at 536 (bg #0 + 536), csum 0x00000000

Inode table at 4129-4640 (bg #0 + 4129)

32255 free blocks, 8192 free inodes, 0 directories, 8192 unused inodes

Free blocks: 229889-262143

Free inodes: 57345-65536

Group 8: (Blocks 262144-294911) csum 0x9bfa [INODE\_UNINIT, BLOCK\_UNINIT, ITABLE\_ZEROED]

Block bitmap at 521 (bg #0 + 521), csum 0x00000000

Inode bitmap at 537 (bg #0 + 537), csum 0x00000000

Inode table at 4641-5152 (bg #0 + 4641)

32768 free blocks, 8192 free inodes, 0 directories, 8192 unused inodes

Free blocks: 262144-294911

Free inodes: 65537-73728

Group 9: (Blocks 294912-327679) csum 0xf3ad [INODE\_UNINIT, BLOCK\_UNINIT, ITABLE\_ZEROED]  
 Backup superblock at 294912, Group descriptors at 294913-294913  
 Reserved GDT blocks at 294914-295424  
 Block bitmap at 522 (bg #0 + 522), csum 0x00000000  
 Inode bitmap at 538 (bg #0 + 538), csum 0x00000000  
 Inode table at 5153-5664 (bg #0 + 5153)  
 32255 free blocks, 8192 free inodes, 0 directories, 8192 unused inodes  
 Free blocks: 295425-327679  
 Free inodes: 73729-81920

Group 10: (Blocks 327680-360447) csum 0xd15d [INODE\_UNINIT, BLOCK\_UNINIT, ITABLE\_ZEROED]  
 Block bitmap at 523 (bg #0 + 523), csum 0x00000000  
 Inode bitmap at 539 (bg #0 + 539), csum 0x00000000  
 Inode table at 5665-6176 (bg #0 + 5665)  
 32768 free blocks, 8192 free inodes, 0 directories, 8192 unused inodes  
 Free blocks: 327680-360447  
 Free inodes: 81921-90112

Group 11: (Blocks 360448-393215) csum 0xc805 [INODE\_UNINIT, BLOCK\_UNINIT, ITABLE\_ZEROED]  
 Block bitmap at 524 (bg #0 + 524), csum 0x00000000  
 Inode bitmap at 540 (bg #0 + 540), csum 0x00000000  
 Inode table at 6177-6688 (bg #0 + 6177)  
 32768 free blocks, 8192 free inodes, 0 directories, 8192 unused inodes  
 Free blocks: 360448-393215  
 Free inodes: 90113-98304

Group 12: (Blocks 393216-425983) csum 0x0eb4 [INODE\_UNINIT, BLOCK\_UNINIT, ITABLE\_ZEROED]  
 Block bitmap at 525 (bg #0 + 525), csum 0x00000000  
 Inode bitmap at 541 (bg #0 + 541), csum 0x00000000  
 Inode table at 6689-7200 (bg #0 + 6689)  
 32768 free blocks, 8192 free inodes, 0 directories, 8192 unused inodes  
 Free blocks: 393216-425983  
 Free inodes: 98305-106496

Group 13: (Blocks 425984-458751) csum 0x124e [INODE\_UNINIT, BLOCK\_UNINIT, ITABLE\_ZEROED]  
 Block bitmap at 526 (bg #0 + 526), csum 0x00000000  
 Inode bitmap at 542 (bg #0 + 542), csum 0x00000000  
 Inode table at 7201-7712 (bg #0 + 7201)  
 32768 free blocks, 8192 free inodes, 0 directories, 8192 unused inodes  
 Free blocks: 425984-458751  
 Free inodes: 106497-114688

Group 14: (Blocks 458752-491519) csum 0x01e3 [INODE\_UNINIT, ITABLE\_ZEROED]  
 Block bitmap at 527 (bg #0 + 527), csum 0x6aeb8f8b  
 Inode bitmap at 543 (bg #0 + 543), csum 0x00000000  
 Inode table at 7713-8224 (bg #0 + 7713)  
 16384 free blocks, 8192 free inodes, 0 directories, 8192 unused inodes  
 Free blocks: 475136-491519  
 Free inodes: 114689-122880

Group 15: (Blocks 491520-524287) csum 0xa7ab [INODE\_UNINIT, BLOCK\_UNINIT, ITABLE\_ZEROED]  
 Block bitmap at 528 (bg #0 + 528), csum 0x00000000  
 Inode bitmap at 544 (bg #0 + 544), csum 0x00000000  
 Inode table at 8225-8736 (bg #0 + 8225)  
 32768 free blocks, 8192 free inodes, 0 directories, 8192 unused inodes  
 Free blocks: 491520-524287  
 Free inodes: 122881-131072

Group 16: (Blocks 524288-557055) csum 0x7469 [ITABLE\_ZEROED]  
 Block bitmap at 524288 (+0), csum 0xeca59ce3  
 Inode bitmap at 524304 (+16), csum 0x3c097576  
 Inode table at 524320-524831 (+32)  
 24543 free blocks, 8189 free inodes, 1 directories, 8187 unused inodes  
 Free blocks: 532513-557055  
 Free inodes: 131074, 131077-139264

Group 17: (Blocks 557056-589823) csum 0x454d [INODE\_UNINIT, ITABLE\_ZEROED]  
 Block bitmap at 524289 (bg #16 + 1), csum 0x0119b0b4  
 Inode bitmap at 524305 (bg #16 + 17), csum 0x00000000  
 Inode table at 524832-525343 (bg #16 + 544)  
 32766 free blocks, 8192 free inodes, 0 directories, 8192 unused inodes  
 Free blocks: 557056, 557059-589823  
 Free inodes: 139265-147456

Group 18: (Blocks 589824-622591) csum 0xc467 [INODE\_UNINIT, BLOCK\_UNINIT, ITABLE\_ZEROED]  
 Block bitmap at 524290 (bg #16 + 2), csum 0x00000000  
 Inode bitmap at 524306 (bg #16 + 18), csum 0x00000000  
 Inode table at 525344-525855 (bg #16 + 1056)  
 32768 free blocks, 8192 free inodes, 0 directories, 8192 unused inodes  
 Free blocks: 589824-622591  
 Free inodes: 147457-155648

Group 19: (Blocks 622592-655359) csum 0xda4c [INODE\_UNINIT, BLOCK\_UNINIT, ITABLE\_ZEROED]  
 Block bitmap at 524291 (bg #16 + 3), csum 0x00000000  
 Inode bitmap at 524307 (bg #16 + 19), csum 0x00000000  
 Inode table at 525856-526367 (bg #16 + 1568)  
 32768 free blocks, 8192 free inodes, 0 directories, 8192 unused inodes  
 Free blocks: 622592-655359  
 Free inodes: 155649-163840

Group 20: (Blocks 655360-688127) csum 0x1b8e [INODE\_UNINIT, BLOCK\_UNINIT, ITABLE\_ZEROED]  
 Block bitmap at 524292 (bg #16 + 4), csum 0x00000000  
 Inode bitmap at 524308 (bg #16 + 20), csum 0x00000000  
 Inode table at 526368-526879 (bg #16 + 2080)  
 32768 free blocks, 8192 free inodes, 0 directories, 8192 unused inodes  
 Free blocks: 655360-688127  
 Free inodes: 163841-172032

Group 21: (Blocks 688128-720895) csum 0x05a5 [INODE\_UNINIT, BLOCK\_UNINIT, ITABLE\_ZEROED]  
 Block bitmap at 524293 (bg #16 + 5), csum 0x00000000  
 Inode bitmap at 524309 (bg #16 + 21), csum 0x00000000  
 Inode table at 526880-527391 (bg #16 + 2592)  
 32768 free blocks, 8192 free inodes, 0 directories, 8192 unused inodes  
 Free blocks: 688128-720895  
 Free inodes: 172033-180224

Group 22: (Blocks 720896-753663) csum 0x5129 [INODE\_UNINIT, BLOCK\_UNINIT, ITABLE\_ZEROED]  
 Block bitmap at 524294 (bg #16 + 6), csum 0x00000000  
 Inode bitmap at 524310 (bg #16 + 22), csum 0x00000000  
 Inode table at 527392-527903 (bg #16 + 3104)  
 32768 free blocks, 8192 free inodes, 0 directories, 8192 unused inodes  
 Free blocks: 720896-753663

```

Free inodes: 180225-188416
Group 23: (Blocks 753664-786431) csum 0x4f02 [INODE_UNINIT, BLOCK_UNINIT, ITABLE_ZEROED]
Block bitmap at 524295 (bg #16 + 7), csum 0x00000000
Inode bitmap at 524311 (bg #16 + 23), csum 0x00000000
Inode table at 527904-528415 (bg #16 + 3616)
32768 free blocks, 8192 free inodes, 0 directories, 8192 unused inodes
Free blocks: 753664-786431
Free inodes: 188417-196608
Group 24: (Blocks 786432-819199) csum 0xd2ad [INODE_UNINIT, BLOCK_UNINIT, ITABLE_ZEROED]
Block bitmap at 524296 (bg #16 + 8), csum 0x00000000
Inode bitmap at 524312 (bg #16 + 24), csum 0x00000000
Inode table at 528416-528927 (bg #16 + 4128)
32768 free blocks, 8192 free inodes, 0 directories, 8192 unused inodes
Free blocks: 786432-819199
Free inodes: 196609-204800
Group 25: (Blocks 819200-851967) csum 0xb82b [INODE_UNINIT, BLOCK_UNINIT, ITABLE_ZEROED]
Backup superblock at 819200, Group descriptors at 819201-819201
Reserved GDT blocks at 819202-819712
Block bitmap at 524297 (bg #16 + 9), csum 0x00000000
Inode bitmap at 524313 (bg #16 + 25), csum 0x00000000
Inode table at 528928-529439 (bg #16 + 4640)
32255 free blocks, 8192 free inodes, 0 directories, 8192 unused inodes
Free blocks: 819713-851967
Free inodes: 204801-212992
Group 26: (Blocks 851968-884735) csum 0x980a [INODE_UNINIT, BLOCK_UNINIT, ITABLE_ZEROED]
Block bitmap at 524298 (bg #16 + 10), csum 0x00000000
Inode bitmap at 524314 (bg #16 + 26), csum 0x00000000
Inode table at 529440-529951 (bg #16 + 5152)
32768 free blocks, 8192 free inodes, 0 directories, 8192 unused inodes
Free blocks: 851968-884735
Free inodes: 212993-221184
Group 27: (Blocks 884736-917503) csum 0xf28c [INODE_UNINIT, BLOCK_UNINIT, ITABLE_ZEROED]
Backup superblock at 884736, Group descriptors at 884737-884737
Reserved GDT blocks at 884738-885248
Block bitmap at 524299 (bg #16 + 11), csum 0x00000000
Inode bitmap at 524315 (bg #16 + 27), csum 0x00000000
Inode table at 529952-530463 (bg #16 + 5664)
32255 free blocks, 8192 free inodes, 0 directories, 8192 unused inodes
Free blocks: 885249-917503
Free inodes: 221185-229376
Group 28: (Blocks 917504-950271) csum 0x47e3 [INODE_UNINIT, BLOCK_UNINIT, ITABLE_ZEROED]
Block bitmap at 524300 (bg #16 + 12), csum 0x00000000
Inode bitmap at 524316 (bg #16 + 28), csum 0x00000000
Inode table at 530464-530975 (bg #16 + 6176)
32768 free blocks, 8192 free inodes, 0 directories, 8192 unused inodes
Free blocks: 917504-950271
Free inodes: 229377-237568
Group 29: (Blocks 950272-983039) csum 0x59c8 [INODE_UNINIT, BLOCK_UNINIT, ITABLE_ZEROED]
Block bitmap at 524301 (bg #16 + 13), csum 0x00000000
Inode bitmap at 524317 (bg #16 + 29), csum 0x00000000
Inode table at 530976-531487 (bg #16 + 6688)
32768 free blocks, 8192 free inodes, 0 directories, 8192 unused inodes
Free blocks: 950272-983039
Free inodes: 237569-245760
Group 30: (Blocks 983040-1015807) csum 0x0d44 [INODE_UNINIT, BLOCK_UNINIT, ITABLE_ZEROED]
Block bitmap at 524302 (bg #16 + 14), csum 0x00000000
Inode bitmap at 524318 (bg #16 + 30), csum 0x00000000
Inode table at 531488-531999 (bg #16 + 7200)
32768 free blocks, 8192 free inodes, 0 directories, 8192 unused inodes
Free blocks: 983040-1015807
Free inodes: 245761-253952
Group 31: (Blocks 1015808-1048232) csum 0x7dd9 [INODE_UNINIT, ITABLE_ZEROED]
Block bitmap at 524303 (bg #16 + 15), csum 0x668eda73
Inode bitmap at 524319 (bg #16 + 31), csum 0x00000000
Inode table at 532000-532511 (bg #16 + 7712)
32425 free blocks, 8192 free inodes, 0 directories, 8192 unused inodes
Free blocks: 1015808-1048232
Free inodes: 253953-262144

```

# mke2fs -n /dev/sda1 [#Amosar a lista dos superbloques, é dicir cales son os bloques que gardan a información do superbloque en /dev/sda1](#)

```

mke2fs 1.43.4 (31-Jan-2017)
/dev/sda1 contains a ext4 file system labelled 'PARTICION1'
last mounted on /mnt/recuperacion on Sun Nov  4 18:47:07 2018
Proceed anyway? (y,N) y
Creating filesystem with 1048233 4k blocks and 262144 inodes
Filesystem UUID: 92a78de8-2f5c-4a20-8329-382c541336ef
Superblock backups stored on blocks:
32768, 98304, 163840, 229376, 294912, 819200, 884736

```

# dumpe2fs /dev/sda2 [#Amosar información sobre o superbloque e grupos de bloques da partición /dev/sda2](#)

```

dumpe2fs 1.43.4 (31-Jan-2017)
Filesystem volume name:  PARTICION2
Last mounted on:
Filesystem UUID:      8d34cbd1-6a9e-4b04-be27-0c29201857b9
Filesystem magic number: 0xEF53
Filesystem revision #: 1 (dynamic)
Filesystem features:  has_journal ext_attr resize_inode dir_index filetype sparse_super large_file
Filesystem flags:      signed_directory_hash
Default mount options: user_xattr acl
Filesystem state:      clean
Errors behavior:       Continue
Filesystem OS type:    Linux
Inode count:           105040
Block count:           419698
Reserved block count:  20984

```

Free blocks: 404276  
Free inodes: 105029  
First block: 0  
Block size: 4096  
Fragment size: 4096  
Reserved GDT blocks: 102  
Blocks per group: 32768  
Fragments per group: 32768  
Inodes per group: 8080  
Inode blocks per group: 505  
Filesystem created: Thu Nov 1 21:22:45 2018  
Last mount time: Thu Nov 1 23:09:58 2018  
Last write time: Fri Nov 2 05:25:02 2018  
Mount count: 2  
Maximum mount count: -1  
Last checked: Thu Nov 1 21:22:45 2018  
Check interval: 0 ()  
Lifetime writes: 72 kB  
Reserved blocks uid: 0 (user root)  
Reserved blocks gid: 0 (group root)  
First inode: 11  
Inode size: 256  
Required extra isize: 32  
Desired extra isize: 32  
Journal inode: 8  
Default directory hash: half\_md4  
Directory Hash Seed: 1b2d810f-9531-4e47-ba6a-27207ca28764  
Journal backup: inode blocks  
Journal features: (none)  
Journal size: 32M  
Journal length: 8192  
Journal sequence: 0x00000006  
Journal start: 0

Group 0: (Blocks 0-32767)  
Primary superblock at 0, Group descriptors at 1-1  
Reserved GDT blocks at 2-103  
Block bitmap at 104 (+104)  
Inode bitmap at 105 (+105)  
Inode table at 106-610 (+106)  
23950 free blocks, 8069 free inodes, 2 directories  
Free blocks: 8818-32767  
Free inodes: 12-8080  
Group 1: (Blocks 32768-65535)  
Backup superblock at 32768, Group descriptors at 32769-32769  
Reserved GDT blocks at 32770-32871  
Block bitmap at 32872 (+104)  
Inode bitmap at 32873 (+105)  
Inode table at 32874-33378 (+106)  
32157 free blocks, 8080 free inodes, 0 directories  
Free blocks: 33379-65535  
Free inodes: 8081-16160  
Group 2: (Blocks 65536-98303)  
Block bitmap at 65536 (+0)  
Inode bitmap at 65537 (+1)  
Inode table at 65538-66042 (+2)  
32261 free blocks, 8080 free inodes, 0 directories  
Free blocks: 66043-98303  
Free inodes: 16161-24240  
Group 3: (Blocks 98304-131071)  
Backup superblock at 98304, Group descriptors at 98305-98305  
Reserved GDT blocks at 98306-98407  
Block bitmap at 98408 (+104)  
Inode bitmap at 98409 (+105)  
Inode table at 98410-98914 (+106)  
32157 free blocks, 8080 free inodes, 0 directories  
Free blocks: 98915-131071  
Free inodes: 24241-32320  
Group 4: (Blocks 131072-163839)  
Block bitmap at 131072 (+0)  
Inode bitmap at 131073 (+1)  
Inode table at 131074-131578 (+2)  
32261 free blocks, 8080 free inodes, 0 directories  
Free blocks: 131579-163839  
Free inodes: 32321-40400  
Group 5: (Blocks 163840-196607)  
Backup superblock at 163840, Group descriptors at 163841-163841  
Reserved GDT blocks at 163842-163943  
Block bitmap at 163944 (+104)  
Inode bitmap at 163945 (+105)  
Inode table at 163946-164450 (+106)  
32157 free blocks, 8080 free inodes, 0 directories  
Free blocks: 164451-196607  
Free inodes: 40401-48480  
Group 6: (Blocks 196608-229375)  
Block bitmap at 196608 (+0)  
Inode bitmap at 196609 (+1)  
Inode table at 196610-197114 (+2)  
32261 free blocks, 8080 free inodes, 0 directories  
Free blocks: 197115-229375  
Free inodes: 48481-56560  
Group 7: (Blocks 229376-262143)  
Backup superblock at 229376, Group descriptors at 229377-229377  
Reserved GDT blocks at 229378-229479  
Block bitmap at 229480 (+104)  
Inode bitmap at 229481 (+105)  
Inode table at 229482-229986 (+106)  
32157 free blocks, 8080 free inodes, 0 directories  
Free blocks: 229987-262143

```
Free inodes: 56561-64640
Group 8: (Blocks 262144-294911)
Block bitmap at 262144 (+0)
Inode bitmap at 262145 (+1)
Inode table at 262146-262650 (+2)
32261 free blocks, 8080 free inodes, 0 directories
Free blocks: 262651-294911
Free inodes: 64641-72720
Group 9: (Blocks 294912-327679)
Backup superblock at 294912, Group descriptors at 294913-294913
Reserved GDT blocks at 294914-295015
Block bitmap at 295016 (+104)
Inode bitmap at 295017 (+105)
Inode table at 295018-295522 (+106)
32157 free blocks, 8080 free inodes, 0 directories
Free blocks: 295523-327679
Free inodes: 72721-80800
Group 10: (Blocks 327680-360447)
Block bitmap at 327680 (+0)
Inode bitmap at 327681 (+1)
Inode table at 327682-328186 (+2)
32261 free blocks, 8080 free inodes, 0 directories
Free blocks: 328187-360447
Free inodes: 80801-88880
Group 11: (Blocks 360448-393215)
Block bitmap at 360448 (+0)
Inode bitmap at 360449 (+1)
Inode table at 360450-360954 (+2)
32261 free blocks, 8080 free inodes, 0 directories
Free blocks: 360955-393215
Free inodes: 88881-96960
Group 12: (Blocks 393216-419697)
Block bitmap at 393216 (+0)
Inode bitmap at 393217 (+1)
Inode table at 393218-393722 (+2)
25975 free blocks, 8080 free inodes, 0 directories
Free blocks: 393723-419697
Free inodes: 96961-105040
```

# mke2fs -n /dev/sda2 #Amosar a lista dos superbloques, é dicir cales son os bloques que gardan a información do superbloque en /dev/sda2

```
mke2fs 1.43.4 (31-Jan-2017)
/dev/sda2 contains a ext3 file system labelled 'PARTICION2'
last mounted on Thu Nov 1 23:09:58 2018
Proceed anyway? (y,N) y
Creating filesystem with 419698 4k blocks and 105040 inodes
Filesystem UUID: f1dd5a37-95ff-4cfa-930a-c88425cb39b4
Superblock backups stored on blocks:
32768, 98304, 163840, 229376, 294912
```

Revisar o contido dos bloques que referencian a información do superbloque.

# debugfs /dev/sda2 #Executar o comando debugfs sobre a partición primaria /dev/sda2

```
debugfs 1.43.4 (31-Jan-2017)
debugfs: block_dump 32768 #Ver o contido do bloque 32768 que contén o backup do superbloque
0000 509a 0100 7267 0600 f851 0000 342b 0600 P...rg...Q..4+..
0020 459a 0100 0000 0000 0200 0000 0200 0000 E.....
0040 0080 0000 0080 0000 901f 0000 0000 0000 .....
0060 a56e db5b 0000 ffff 53ef 0000 0100 0000 .n.[...S.....
0100 a56e db5b 0000 0000 0000 0000 0100 0000 .n.[.....
0120 0000 0000 0b00 0000 0001 0100 3c00 0000 .....<...
0140 0200 0000 0300 0000 8d34 cbd1 6a9e 4b04 .....4..j.K.
0160 be27 0c29 2018 57b9 5041 5254 4943 494f .'.).W.PARTICIO
0200 4e32 0000 0000 0000 0000 0000 0000 0000 N2.....
0220 0000 0000 0000 0000 0000 0000 0000 0000 .....
*
0300 0000 0000 0000 0000 0000 0000 0000 6600 .....f.
0320 0000 0000 0000 0000 0000 0000 0000 0000 .....
0340 0800 0000 0000 0000 0000 0000 1b2d 810f .....-..
0360 9531 4e47 ba6a 2720 7ca2 8764 0101 0000 .1NG.j' |.d...
0400 0c00 0000 0000 0000 a56e db5b 6902 0000 .....n.[i...
0420 6a02 0000 6b02 0000 6c02 0000 6d02 0000 j...k...l...m...
0440 6e02 0000 6f02 0000 7002 0000 7102 0000 n...o...p...q...
0460 7202 0000 7302 0000 7402 0000 7502 0000 r...s...t...u...
0500 7606 0000 0000 0000 0000 0000 0000 0002 v.....
0520 0000 0000 0000 0000 0000 0000 2000 2000 ..... . .
0540 0100 0000 0000 0000 0000 0000 0000 0000 .....
0560 0000 0000 0000 0000 0000 0000 0000 0000 .....
*
debugfs: q #Premier a tecla q para saír da consola debugfs
```

Ricardo Feijoo Costa



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

# losetup (dispositivos de bloques virtual loop)

## Práctica

```
# mount -o loop file.iso /media/cdrom0 #Montar dispositivo iso en /media/cdrom0.

# cd && umount /media/cdrom0 #Desmontar /media/cdrom0.

# dd if=/dev/zero of=file1.raw bs=1MiB count=100 #Crear un ficheiro file1.raw que contén todos ceros no
directorio actual cun tamaño de 100MiB.

# dd if=/dev/zero of=file2.raw bs=100MiB count=2 #Crear un ficheiro file2.raw que contén todos ceros no
directorio actual cun tamaño de 200MiB.

# losetup -f --show file1.raw #Enlazar a file1.raw o primeiro dispositivo loop libre (-f), e amosando cal é (--show).

# losetup -a #Amosar tódolos dispositivos loop enlazados.

# mount -o loop file.iso /media/cdrom0 #Montar dispositivo iso en /media/cdrom0.

# losetup -a #Amosar tódolos dispositivos loop enlazados.

# cd && umount /media/cdrom0 #Desmontar /media/cdrom0.

# parted --script /dev/loop0 mklabel msdos #Crear a etiqueta de disco ao dispositivo /dev/loop0 sen ter que
acceder ao prompt de parted

# parted --script /dev/loop0 mkpart primary 0 50% #Crear unha partición primaria co primeiro 50% do dispositivo
/dev/loop0 sen ter que acceder ao prompt de parted

# parted --script /dev/loop0 mkpart primary 50% 100% #Crear unha partición primaria co último 50% do
dispositivo /dev/loop0 sen ter que acceder ao prompt de parted

# ls -lah /dev/loop0* #Listar o dispositivo /dev/loop0 e as súas particións (xeradas anteriormente): /dev/loop0p1,
/dev/loop0p2

# mkfs.ext4 -L 'parte1loop0' /dev/loop0p1 #Formatear en ext4 a partición /dev/loop0p1 etiquetada como
parte1loop0.

# mkfs.ext4 -L 'parte2loop0' /dev/loop0p2 #Formatear en ext4 a partición /dev/loop0p2 etiquetada como
parte2loop0.

# mkdir -p /media/loop0/loop0p1 /media/loop0/loop0p2 #Crear cartafol /media/loop0/loop0p1 e
/media/loop0/loop0p2

# mount /dev/loop0p1 /media/loop0/loop0p1 #Montar /dev/loop0p1 en /media/loop0/loop0p1
```



```
# mount /dev/loop0p2 /media/loop0/loop0p2 #Montar /dev/loop0p2 en /media/loop0/loop0p2

# mount | grep loop #Amosar dispositivos montados que concordan co patrón loop

# cp -pv /etc/passwd /media/loop0/loop0p1 #Copiar o ficheiro /etc/passwd en /media/loop0/loop0p1

# umount /media/loop0/loop0p1 /media/loop0/loop0p2 #Desmontar /media/loop0/loop0p1 e /media/loop0/loop0p2

# losetup -d /dev/loop0 #Desmontar e desenlazar o dispositivo /dev/loop0
```

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/)

# RAID por Software: mdadm

## mdadm

```
# apt-cache show mdadm
# echo "mdadm mdadm/initrdstart string none" | debconf-set-selections
# apt-get install mdadm
# debconf-show mdadm
mdadm/autocheck: true
mdadm/initrdstart_msg_errconf:
mdadm/mail_to: root
mdadm/start_daemon: true
mdadm/initrdstart_msg_errmd:
mdadm/initrdstart_notinconf: false
* mdadm/initrdstart: none
mdadm/initrdstart_msg_errblock:
mdadm/initrdstart_msg_intro:
mdadm/initrdstart_msg_errexist:
# man 8 mdadm
# cat /etc/mdadm/mdadm.conf
# cat /proc/mdstat
```

Para facer a práctica imos empregar dispositivos virtuais loop, aínda que poderíamos dispositivos reais, como: /dev/sde1, /dev/sdf4, etc.

## Práctica

### Preparación discos e particións

```
# dd if=/dev/zero of=file1.raw bs=1MiB count=100 #Crear un ficheiro file1.raw que contén todos ceros no
directorio actual cun tamaño de 100MiB.

# dd if=/dev/zero of=file2.raw bs=100MiB count=2 #Crear un ficheiro file2.raw que contén todos ceros no
directorio actual cun tamaño de 200MiB.

# dd if=/dev/zero of=file3.raw bs=1MiB count=100 #Crear un ficheiro file3.raw que contén todos ceros no
directorio actual cun tamaño de 100MiB.

# for i in 1 2 3; do losetup -f --show file${i}.raw;done #Enlazar file1.raw, file2.raw e file3.raw aos primeiros
dispositivos loop libres (-f), amosando cales son (--show).

# losetup -a #Amosar tódolos dispositivos loop enlazados.

# parted --script /dev/loop0 mklabel msdos #Crear a etiqueta de disco ao dispositivo /dev/loop0 sen ter que acceder
ao prompt de parted

# parted --script /dev/loop0 mkpart primary 0 50% 2>/dev/null #Crear unha partición primaria co primeiro 50%
do dispositivo /dev/loop0 sen ter que acceder ao prompt de parted

# parted --script /dev/loop0 mkpart primary 50% 100% 2>/dev/null #Crear unha partición primaria co último
50% do dispositivo /dev/loop0 sen ter que acceder ao prompt de parted

# ls -lah /dev/loop0* #Listar o dispositivo /dev/loop0 e as súas particións (xeradas anteriormente): /dev/loop0p1,
```

```
/dev/loop0p2
```

```
# mkfs.ext4 -L 'parte1loop0' /dev/loop0p1 #Formatear en ext4 a partición /dev/loop0p1 etiquetada como parte1loop0.
```

```
# mkfs.ext4 -L 'parte2loop0' /dev/loop0p2 #Formatear en ext4 a partición /dev/loop0p2 etiquetada como parte2loop0.
```

```
# mkfs.ext4 -L 'fullloop1' /dev/loop1 #Formatear en ext4 o dispositivo /dev/loop1
```

```
# mkfs.ext4 -L 'fullloop2' /dev/loop2 #Formatear en ext4 o dispositivo /dev/loop2
```

## --create: Creación de RAID

```
# yes | mdadm --create /dev/md0 --level=0 --raid-devices=2 /dev/loop0p1 /dev/loop1 #Crear un volume RAID0 en /dev/md0 usando a partición /dev/loop0p1 e o dispositivo /dev/loop1.
```

```
# yes | mdadm --create /dev/md1 --level=1 --raid-devices=2 /dev/loop0p2 /dev/loop2 #Crear un volume RAID1 en /dev/md1 usando a partición /dev/loop0p2 e o dispositivo /dev/loop2.
```

## --query, --detail, /proc/mdstat: Escanear e amosar información RAID

```
# mdadm --query /dev/md0 #Amosa información reducida sobre o volume RAID /dev/md0
```

```
# mdadm --detail /dev/md0 #Amosa información extendida sobre o volume RAID /dev/md0
```

```
# cat /proc/mdstat #Amosa información sobre o estado actual do volume RAID
```

## (--fail, --remove)Quitar/Engadir(--add) discos ao RAID

```
# mdadm --fail /dev/md1 /dev/loop2 #Para quitar o dispositivo /dev/loop2 do RAID, éste debe estar en estado fallido (fail), co cal provocamos este estado para poder quitalo
```

```
# cat /proc/mdstat
```

```
# mdadm --remove /dev/md1 /dev/loop2 #Quitamos o dispositivo fallido /dev/loop2 do RAID
```

```
# cat /proc/mdstat
```

```
# mdadm --add /dev/md1 /dev/loop2 #Engadir dispositivo /dev/loop2 do RAID
```

## Formatear o RAID

```
# mkfs.ext4 -L 'RAID0' /dev/md0 #Formatear en ext4 o volume RAID0 etiquetado como RAID0.
```

```
# mkfs.ext4 -L 'RAID1' /dev/md1 #Formatear en ext4 o volume RAID1 etiquetado como RAID1.
```

## Montar RAID

```
# mkdir -p /media/raid0 && mkdir -p /media/raid1 #Crear os cartafolés /media/raid0 e /media/raid1

# mount /dev/md0 /media/raid0 #Montar o volume RAID0 en /media/raid0

# mount /dev/md1 /media/raid1 #Montar o volume RAID1 en /media/raid1

# mount | grep raid #Amosar dispositivos montados que concordan co patrón raid

# cp -pv /etc/passwd /media/raid0 #Copiar o ficheiro /etc/passwd en /media/raid0

# df -h | grep raid0 #Ver a utilización do espazo en disco do volume RAID0

# umount /media/raid0 /media/raid1 #Desmontar /media/raid0 e /media/raid1
```

## --stop, --remove, --zero-superblock: Eliminación de volumes RAID

```
# mdadm --stop /dev/md0 #Asegurarse de parar o volume RAID0 antes de eliminalo

# mdadm --remove /dev/md0 #Eliminar volume RAID0

# mdadm --zero-superblock /dev/loop0p1 /dev/loop1 #Liberar a asociación de dispositivos ao volume RAID0

# mdadm --stop /dev/md1 #Asegurarse de parar o volume RAID1 antes de eliminalo

# mdadm --remove /dev/md1 #Eliminar volume RAID1

# mdadm --zero-superblock /dev/loop0p2 /dev/loop2 #Liberar a asociación de dispositivos ao volume RAID1
```

## losetup -d : Desconectar dispositivos loop

```
# for i in 0 1 2; do losetup -d /dev/loop${i};done #Desmontar e desenlazar os dispositivos /dev/loop0, /dev/loop1 e /dev/loop2

# losetup -a #Amosar tódolos dispositivos loop enlazados.
```

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/)

# LVM: physical volume(pvX), volume group (vgX), logical volume(lvX)

## lvm2 (pvX, vgX, lvX)

```
# apt-cache show lvm2
# apt-get install lvm2
# man 8 lvm
```

Para facer a práctica imos empregar dispositivos virtuais loop, aínda que poderíamos empregar comandos coma os seguintes con dispositivos reais:

```
# mkfs.ext4 -L 'parte1' /dev/sde1 #Formatear en ext4 /dev/sde1 etiquetado como parte1.
# mkfs.ext4 -L 'parte2' /dev/sdf4 #Formatear en ext4 /dev/sdf4 etiquetado como parte2.
# mkfs.ext4 -L 'parte3' /dev/sdg #Formatear en ext4 /dev/sdg etiquetado como parte3.
# pvcreate /dev/sde1 /dev/sdf4 /dev/sdg #Inicializar as particións /dev/sda1, /dev/sdf4 e o disco /dev/sdg para usar por LVM.
# vgcreate primeiro_vg /dev/sde1 /dev/sdf4 /dev/sdg #Crea un grupo de volumes usando as particións /dev/sda1, /dev/sdf4 e o disco /dev/sdg.
# lvcreate -L 20G -n primeiro_lv primeiro_vg #Crear un volume lóxico dentro do grupo de volumes primeiro_vg de 20G denominado primeiro_lv
# mkfs.ext4 -L 'volumeOK' /dev/primeiro_vg/primeiro_lv #Formatear en ext4 o volume lóxico primeiro_lv etiquetado como volumeOK.
```

## Práctica

### Preparación discos e particións

```
# dd if=/dev/zero of=file1.raw bs=1MiB count=100 #Crear un ficheiro file1.raw que contén todos ceros no directorio actual cun tamaño de 100MiB.

# dd if=/dev/zero of=file2.raw bs=100MiB count=2 #Crear un ficheiro file2.raw que contén todos ceros no directorio actual cun tamaño de 200MiB.

# dd if=/dev/zero of=file3.raw bs=1MiB count=100 #Crear un ficheiro file3.raw que contén todos ceros no directorio actual cun tamaño de 100MiB.

# for i in 1 2 3; do losetup -f --show file${i}.raw;done #Enlazar file1.raw, file2.raw e file3.raw aos primeiros dispositivos loop libres (-f), amosando cales son (--show).

# losetup -a #Amosar tódolos dispositivos loop enlazados.

# parted --script /dev/loop0 mklabel msdos #Crear a etiqueta de disco ao dispositivo /dev/loop0 sen ter que acceder ao prompt de parted

# parted --script /dev/loop0 mkpart primary 0 50% 2>/dev/null #Crear unha partición primaria co primeiro 50% do dispositivo /dev/loop0 sen ter que acceder ao prompt de parted

# parted --script /dev/loop0 mkpart primary 50% 100% 2>/dev/null #Crear unha partición primaria co último 50% do dispositivo /dev/loop0 sen ter que acceder ao prompt de parted

# ls -lah /dev/loop0* #Listar o dispositivo /dev/loop0 e as súas particións (xeradas anteriormente): /dev/loop0p1,
```

```
/dev/loop0p2
```

```
# mkfs.ext4 -L 'parte1loop0' /dev/loop0p1 #Formatear en ext4 a partición /dev/loop0p1 etiquetada como parte1loop0.
```

```
# mkfs.ext4 -L 'parte2loop0' /dev/loop0p2 #Formatear en ext4 a partición /dev/loop0p2 etiquetada como parte2loop0.
```

```
# mkfs.ext4 -L 'fullloop1' /dev/loop1 #Formatear en ext4 o dispositivo /dev/loop1
```

```
# mkfs.ext4 -L 'fullloop2' /dev/loop2 #Formatear en ext4 o dispositivo /dev/loop2
```

## **pvcreate, vgcreate, lvcreate: Creación de volumes lóxicos**

```
# pvcreate /dev/loop0p1 /dev/loop1 /dev/loop2 #Inicializar as particións /dev/loop0p1 e os dispositivos /dev/loop1, /dev/loop2 para usar por LVM.
```

```
# vgcreate segundo_vg /dev/loop0p1 /dev/loop1 /dev/loop2 #Crea un grupo de volumes usando as particións /dev/loop0p1 e os dispositivos /dev/loop1, /dev/loop2
```

```
# lvcreate -L 200M -n segundo_lv segundo_vg #Crear un volume lóxico dentro do grupo de volumes segundo_vg de 200M denominado segundo_lv
```

## **Escanear e amosar información LVM**

```
# pvscan; vgscan; lvscan #Permiten escanear respectivamente: volumes físicos, grupos de volumes e volumes lóxicos existentes.
```

```
# pvdisplay; vgdisplay; lvdisplay #Amosa respectivamente información das propiedades dos volumes físicos, grupos de volumes e volumes lóxicos.
```

```
# pvs; vgs; lvs #Amosa respectivamente información resumida sobre volumes físicos, grupos de volumes e volumes lóxicos.
```

## **Activar/Desactivar LVM**

```
# vgchange -aay #Activa tódolos grupos de volumes: escanea os dispositivos dispoñibles e inicialízaos.
```

```
# vgchange -an segundo_vg #Desactiva o grupo de volume segundo_vg.
```

```
# vgchange -ay segundo_vg #Activa o grupo de volume segundo_vg.
```

## **Extender/Reducir Grupo de Volumes**

```
# vgreduce segundo_vg /dev/loop2 #Quitar o dispositivo /dev/loop2 do grupo de volumes segundo_vg
```

```
# pvdisplay
```

```
# vgextend segundo_vg /dev/loop2 #Engadir o dispositivo /dev/loop2 ao grupo de volumes segundo_vg
```

```
# pvdisplay
```

## Formatear e redimensionar LVM

```
# mkfs.ext4 -L '2volumeOK' /dev/segundo_vg/segundo_lv #Formatear en ext4 o volume lóxico segundo_lv etiquetado como 2volumeOK.  
  
# lvresize -L 320M segundo_vg/segundo_lv #Redimensionar a 320M o volume lóxico segundo_lv  
  
# resize2fs /dev/segundo_vg/segundo_lv #Adaptar o sistema de ficheiros ao novo tamaño.
```

## Montar LVM

```
# mkdir -p /media/lvm2 #Crear cartafol /media/lvm2  
  
# mount /dev/segundo_vg/segundo_lv /media/lvm2 #Montar /dev/segundo_vg/segundo_lv en /media/lvm2  
  
# mount | grep lvm #Amosar dispositivos montados que concordan co patrón lvm  
  
# cp -pv /etc/passwd /media/lvm2 #Copiar o ficheiro /etc/passwd en /media/lvm2  
  
# df -h | grep segundo_lv #Ver a utilización do espazo en disco do volume lóxico segundo_lv  
  
# umount /media/lvm2 #Desmontar /media/lvm2
```

## lvremove, vgremove, pvremove: Eliminación de volumes lóxicos

```
# umount /media/lvm2 #Asegurarse en desmontar /media/lvm2  
  
# lvremove /dev/segundo_vg/segundo_lv #Eliminar volume lóxico segundo_lv  
  
# vgremove segundo_vg #Eliminar o grupo de volumes segundo_vg  
  
# pvremove /dev/loop2 #Eliminar o volume físico /dev/loop2
```

## losetup -d : Desconectar dispositivos loop

```
# for i in 0 1 2; do losetup -d /dev/loop${i};done #Desmontar e desenlazar os dispositivos /dev/loop0, /dev/loop1 e /dev/loop2  
  
# losetup -a #Amosar tódolos dispositivos loop enlazados.
```

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/)

# Atopar información sobre o kernel, sistema operativo e hardware:

**uname, dmesg, journalctl, dmidecode, hwinfo, /etc/debian\_version/, lsb\_release, lsbblk, lspci, lspci, lsusb**

- **uname:** Amosa información sobre o kernel e o sistema operativo.

```
$ uname #Amosa o nome do sistema operativo. Igual que uname -s.
```

```
$ uname -r #Amosa o kernel release.
```

```
$ uname -v #Amosa a versión do kernel (data compilación).
```

```
$ uname -a #Amosa toda información que permite o comando.
```

- **dmesg** #Amosa as mensaxes do kernel acontecidas.

```
$ su - #Abrir unha (sub)consola para o usuario root cargando as súas variables de entorno. A nova consola equivale á que tería o usuario cando fai login.
```

```
# dmesg #Amosa as mensaxes do kernel acontecidas dende o arranque do sistema operativo ata a execución do comando.
```

```
# dmesg -w #Amosa as mensaxes do kernel acontecidas e espera a próximas conexións en tempo real sen devolver o prompt. A opción -w é válida dende a versión do kernel 3.5.0
```

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

- **journalctl:** Amosa o rexistro de logs de systemd.

```
$ su - #Abrir unha (sub)consola para o usuario root cargando as súas variables de entorno. A nova consola equivale á que tería o usuario cando fai login.
```

```
# journalctl -f #Amosa as últimas mensaxes referentes ao sistema operativo, é dicir, todo o referente aos contidos de systemd e espera a próximas conexións en tempo real sen devolver o prompt.
```

```
# journalctl -kf #Amosa as mensaxes do kernel acontecidas e espera a próximas conexións en tempo real sen devolver o prompt.
```

```
# journalctl _SYSTEMD_UNIT=ssh.service #Amosa as mensaxes systemd referidas ao servizo ssh. (Ver man systemd.journal-fields)
```

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

- **dmidecode:** Lista información sobre a táboa DMI (SMBIOS) de un ordenador, é dicir, información sobre o hardware do equipo, así como números de serie e revisión de BIOS.

```
$ man dmidecode #Acceder á páxina do manual do comando dmidecode. Ver os tipos existentes (-type)
```

```
$ su - #Abrir unha (sub)consola para o usuario root cargando as súas variables de entorno. A nova consola equivale á que tería o usuario cando fai login.
```

```
# dmidecode #Lista información sobre todo o hardware, versións e BIOS.
```

```
# dmidecode --type #Lista os posibles compoñentes a buscar información.
```

```
# dmidecode --type processor #Lista información sobre o procesador.
```

```
# dmidecode --type 4 #Equivalente ao comando anterior.
```

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

- **hwinfo:** Lista información sobre o hardware do equipo.

```
$ man hwinfo #Acceder á páxina do manual do comando hwinfo. Ver os hardware item existentes (--all)
```

```
$ su - #Abrir unha (sub)consola para o usuario root cargando as súas variables de entorno. A nova consola equivale á que tería o usuario cando fai login.
```



# hwinfo #Lista información sobre todo o hardware do equipo. Equivale a *hwinfo --all --log=-'*

# hwinfo --all #Lista información sobre todo o hardware do equipo.

# hwinfo --cpu #Lista información sobre procesador/es.

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

- /etc/debian\_version: Ficheiro que garda a información sobre a versión do sistema operativo debian.

\$ cat /etc/debian\_version #Ver o contido do ficheiro /etc/debian\_version.

- lsb\_release: Utilidade que informa da versión da base estándar de Linux (LSB).

\$ lsb\_release -a #Mosa toda a información LSB posible: distribuidor, descrición, versión estable e codename.

- lsblk: Lista dispositivos de bloques. Consegue a información do sistema de ficheiros sysfs e a base de datos udev.
- lscpu: Lista información sobre a arquitectura da CPU. Consegue a información do sistema de ficheiros sysfs, do arquivo /proc/cpuinfo e dalguna librería específica da arquitectura.
- lspci: Lista todos os dispositivos PCI.
- lsusb: Lista todos os dispositivos USB.

Ricardo Feijoo Costa



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

# Módulos kernel:

## lspci, lsusb, dmesg, journalctl, lsmod, modprobe (insmod, rmmod), modinfo

- lspci: Lista todos os dispositivos PCI.

\$ lspci -vvv #Lista todos os dispositivos PCI no modo verbose(detallado) máximo.

\$ lspci -k #Lista todos os dispositivos PCI e o driver que emprega cada un no kernel, é dicir, lista tamén os módulos utilizados por cada dispositivo PCI

\$ lspci -k -vvv #Lista todos os dispositivos PCI e o driver que emprega cada un no kernel, é dicir, lista tamén os módulos utilizados por cada dispositivo PCI no modo verbose(detallado) máximo.

\$ lspci -k | grep -A2 -i ethernet #Ver información dos dispositivos PCI ethernet filtrando a saída según o patrón ethernet independente de maiúsculas e minúsculas, e unha vez atopada a referencia amosa as 2 seguintes liñas.

\$ lspci -k | grep -A2 -i ethernet | grep Kernel | cut -d':' -f2 #Listar soamente o/s módulo/s que emprega/n o/s dispositivo/s ethernet. Así, fai o mesmo que o comando anterior, pero a maiores co comando cut amosa soamente a segunda columna do atopado, sendo o campo separador de columnas o carácter ':'

- lsusb: Lista todos os dispositivos USB.

\$ lsusb -v #Lista todos os dispositivos USB no modo verbose(detallado).

\$ lsusb -t #Lista todos os dispositivos USB e o driver que emprega cada un no kernel, é dicir, lista tamén os módulos utilizados por cada dispositivo USB

- dmesg: Amosa mensaxes do kernel dende o arranque, polo cal é útil para observar a carga de módulos no kernel no arranque ou en tempo real.

### Escenario

- Imos probar como responde o kernel ao conectar/desconectar un dispositivo USB. Por exemplo: unha tarxeta de rede sen fíos.

1. Antes de conectar:

\$ dmesg -w #Amosa as mensaxes do kernel acontecidas e espera a próximas conexións en tempo real sen devolver o prompt. A opción -w é válida dende a versión do kernel 3.5.0

2. Conectamos o dispositivo USB.

3. Aparece información sobre a carga deste: Product, Manufacturer, SerialNumber, módulo (driver), firmware, nome interface...

4. Desconectamos o dispositivo USB.

5. Aparece información sobre a descarga deste.

\$ <Ctrl>+<c> #Cancelar comando devolvendo o prompt do sistema.

- journalctl: Amosa o rexistro de logs de systemd.

#### Escenario

- Imos probar como responde o kernel ao conectar/desconectar un dispositivo USB. Por exemplo: unha tarxeta de rede sen fíos.

1. Antes de conectar:

# journalctl -kf #Amosa as mensaxes do kernel acontecidas e espera a próximas conexións en tempo real sen devolver o prompt.

2. Conectamos o dispositivo USB.
3. Aparece información sobre a carga deste: Product, Manufacturer, SerialNumber, módulo (driver), firmware, nome interface...
4. Desconectamos o dispositivo USB.
5. Aparece información sobre a descarga deste.

# <Ctrl>+<c> #Cancelar comando devolvendo o prompt do sistema.

- lsmod: Listar o estado dos módulos no kernel. Recolle a información do arquivo /proc/modules e amosa a información de forma máis lexible.

\$ lsmod | less #Permite moverse mediante o comando less sobre a información amosada polo comando lsmod

- modprobe: Carga ou descarga módulos do kernel Linux.

#### Escenario:

- Interface ethernet: eth0
- Módulo kernel interface eth0: e1000
- ifconfig amosa información de rede sobre eth0

# modprobe -r e1000 #Descargar o módulo do kernel e1000 para a interface eth0. Así, a tarxeta eth0 deixa de estar activa no sistema, polo que o comando ifconfig ou ifconfig -a non amosa a tarxeta cargada no sistema.

# modprobe e1000 #Carga o módulo do kernel e1000 para a interface eth0. Así, a tarxeta eth0 actívase no sistema, polo que o comando ifconfig amosa a tarxeta cargada no sistema.

- modinfo: Amosa información sobre os módulos do kernel. Se e1000 é o nome dun módulo e non un ficheiro a información búscase en /lib/modules/\$(uname -r)

\$ modinfo e1000 #Amosa información sobre o módulo e1000.

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/)

# hotplug and udev



**/dev, udev, udevadm, /etc/udev/rules.d/**

- /dev: Ficheiros de dispositivos de bloques e caracteres do sistema operativo.
- udev: Xestión dinámica de dispositivos /dev
- udevadm: Ferramenta de xestión de udev
- /etc/udev/rules.d: Regras udev para modificación/creación por parte de root. Outras zonas de regras udev: /lib/udev/rules.d (directorio de regras do sistema), /run/udev/rules.d (directorio de regras en tempo de execución volátil).

## Práctica

### 1. Abrir un terminal:

```
$ su - #Abrir unha (sub)consola para o usuario root cargando as súas variables de entorno. Similar a facer login.  
# dmesg -w #Amosa as mensaxes do kernel acontecidas e espera a próximas conexións en tempo real sen devolver o prompt. A opción -w é válida dende a versión do kernel 3.5.0
```

### 2. Conectar un dispositivo de bloques USB (pendrive). Podemos ver na consola a identificación deste dispositivo conectado(sdb):

```
[27629.051850] usb 2-3: new SuperSpeed Gen 1 USB device number 18 using xhci_hcd  
[27629.073362] usb 2-3: New USB device found, idVendor=0951, idProduct=1666, bcdDevice= 1.10  
[27629.073367] usb 2-3: New USB device strings: Mfr=1, Product=2, SerialNumber=3  
[27629.073370] usb 2-3: Product: DataTraveler 3.0  
[27629.073373] usb 2-3: Manufacturer: Kingston  
[27629.073376] usb 2-3: SerialNumber: 1C6F654E4168B28089147521  
[27629.075337] usb-storage 2-3:1.0: USB Mass Storage device detected  
[27629.076012] scsi host1: usb-storage 2-3:1.0  
[27630.101835] scsi 1:0:0:0: Direct-Access Kingston DataTraveler 3.0 PMAP PQ: 0 ANSI: 6  
[27630.102550] sd 1:0:0:0: Attached scsi generic sgl type 0  
[27630.103442] sd 1:0:0:0: [sdb] 30277632 512-byte logical blocks: (15.5 GB/14.4 GiB)  
[27630.104004] sd 1:0:0:0: [sdb] Write Protect is off  
[27630.104011] sd 1:0:0:0: [sdb] Mode Sense: 45 00 00 00  
[27630.104579] sd 1:0:0:0: [sdb] Write cache: disabled, read cache: enabled, doesn't support DPO or FUA  
[27630.154374] sdb: sdb1 sdb2  
[27630.158187] sd 1:0:0:0: [sdb] Attached SCSI removable disk
```

### 3. Abrir outro terminal:

```
$ udevadm info -a -n /dev/sdb #Amosar información udev do dispositivo de bloques /dev/sdb  
$ udevadm info -a -n sdb #Equivale ao comando anterior
```

### 4. Imos crear unha regra udev en /etc/udev/rules.d, que identifique o dispositivo sdb anterior para executar un script bash:

```
$ cat > /usr/bin/hotplug.sh << EOF #Crear o script bash  
> #!/bin/bash  
> export DISPLAY=:0  
> su - -c 'startx -- :10 &'  
> su - -c 'terminator --display=:10 &'  
> su - usuario -c 'terminator --display=:0'  
> su - usuario -c 'firefox --display=:0 www.google.es'  
EOF  
  
$ chmod +x /usr/bin/hotplug.sh #Otorgar permiso de execución ao ficheiro /usr/bin/hotplug.sh  
  
$ su - #Abrir unha (sub)consola para o usuario root cargando as súas variables de entorno. A nova consola equivale á que tería o usuario cando fai login.  
  
# echo 'ACTION=="add", KERNEL=="sd*", ATTR{size}=="30277632",  
ATTRS{vendor}=="Kingston", ATTRS{model}=="DataTraveler 3.0", RUN+="/usr/bin/hotplug.sh"  
>> /etc/udev/rules.d/99-pendrive-usb.rules
```

### 5. Desconectar o pendrive USB /dev/sdb

### 6. Voltar a conectar o pendrive USB /dev/sdb. Agora executarase o script /usr/bin/hotplug.sh

Ricardo Feijoo Costa



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

# Permisos SUID (4000 ou S ou s no usuario propietario)

## S cando non posúe permiso de execución

## s cando posúe permiso de execución

**ESCENARIO:** Para verificar o que acontece na práctica crear en VirtualBox a seguinte máquina virtual:

1. Nome: **Debian32-Permisos**
2. ISO Live Debian 32bits, escritorio XFCE
3. Sen disco duro virtual.
4. RAM: 1024MB
5. Número de procesadores: 2.
6. Primeira opción de arranque: CD Virtual.
7. Rede: Modo Rede Interna
8. No xestor de arranque escoller a primeira opción (opción por defecto).

### Práctica

\$ whoami #Amosa con que usuario estou a traballar na consola

user

\$ find /usr/bin/p\* -type f -perm -4000 -exec ls -l {} \; #Atopar dentro da ruta **/usr/bin** todos os arquivos nos que o nome comeza por **p** e logo posúa ningún carácter ou calquera número de caracteres, e posúan o permiso **SUID(4000)** para logo facerlle un listado extendido mediante **ls -l**

```
-rwsr-xr-x 1 root root 59680 Mai 17 2017 /usr/bin/passwd
-rwsr-xr-x 1 root root 23352 Dec 6 18:34 /usr/bin/pkexec
-rwsr-xr-- 1 root plugdev 49496 Mai 18 2014 /usr/bin/pmount
-rwsr-sr-x 1 root mail 93456 Nov 19 2017 /usr/bin/procmail
-rwsr-xr-- 1 root plugdev 35896 Mai 18 2014 /usr/bin/pumount
```

Vemos que o comando **passwd (/usr/bin/passwd)**, o cal permite xerar/modificar o contrasinal dos usuarios, ten o **bit SUID activado** sendo o seu usuario propietario o usuario **root**, co cal, cada vez que se executa este comando por calquera usuario, os permisos cos cales se está a executar son os permisos de **root**, é dicir, durante a execución deste comando somos o usuario **root** cos permisos efectivos **rwX**.

\$ ls -l /etc/passwd #Listar de forma extendida o ficheiro **/etc/passwd**

```
-rw-r--r-- 1 root root 2490 Mar 22 10:53 /etc/passwd
```

\$ cat /etc/passwd #Ver a información sobre os usuarios do sistema

\$ ls -l /etc/shadow #Listar de forma extendida o ficheiro **/etc/shadow**

```
-rw-r----- 1 root shadow 1593 Mar 22 10:53 /etc/shadow
```

\$ cat /etc/shadow #Ver a información sobre os contrasinais do sistema

cat: /etc/shadow: Permiso denegado

Non se pode amosar o contido dese ficheiro xa que o usuario **user**, co cal, estamos a traballar, non posúe ningún permiso sobre ese ficheiro, xa que na máscara de permisos posúe os permisos **---** correspondentes aos permisos **outros** dos permisos **ugo**

Entón como é posible que poidamos modificar o ficheiro **/etc/shadow** cando cambiamos o contrasinal do usuario? É dicir, como o usuario **user** pode modificar o ficheiro **/etc/shadow** cando actualiza o seu contrasinal? Pois, mediante o permiso **SUID** do comando **passwd**. Debido a que cando executa este comando posúe permisos efectivos de **root**, neste caso **rw** correspondentes aos permisos do usuario propietario dos permisos **ugo**, podendo así modificar o ficheiro **/etc/shadow**.

\$ passwd #Modificar o contrasinal do usuario co cal estou a traballar, neste caso o usuario **user**

```
Changing password for user.
(current) UNIX password:
Enter new UNIX password:
Retype new UNIX password:
passwd: o contrasinal actualizouse con éxito
```

\$ ls -l /etc/shadow #Listar de forma extendida o ficheiro **/etc/shadow**

```
-rw-r----- 1 root shadow 1593 Abr 23 20:18 /etc/shadow
```

\$ cat /etc/shadow #Ver a información sobre os contrasinais do sistema

cat: /etc/shadow: Permiso denegado

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/)

# Permisos Sticky bit (1000 ou T ou t nos permisos outros de ugo)

T cando non posúe permiso de execución

t cando posúe permiso de execución

**ESCENARIO:** Para verificar o que acontece na práctica crear en VirtualBox a seguinte máquina virtual:

1. Nome: **Debian32-Permisos**
2. ISO Live Debian 32bits, escritorio XFCE
3. Sen disco duro virtual.
4. RAM: 1024MB
5. Número de procesadores: 2.
6. Primeira opción de arranque: CD Virtual.
7. Rede: Modo Rede Interna
8. No xestor de arranque escoller a primeira opción (opción por defecto).

## Práctica

\$ whoami #Amosa con que usuario estou a traballar na consola

user

\$ ls -ld /tmp #Listar de forma extendida os permisos do directorio /tmp

drwxrwxrwt 18 root root 15360 Abr 23 21:47 /tmp

\$ sudo su - #Acceder como usuario **root** cargando os arquivos de configuración deste usuario, é dicir, cargando unha contorna similar á que tería o usuario facendo **login** directamente na consola.

\$ useradd -m -s /bin/bash -d /home/user1 -p \$(mkpasswd -m sha-256 abc123.) user1 #Crear o usuario de nome **user1**, onde:

-m crea o directorio **/home/user1** se este non existe, e ese directorio conterá o predefinido en **/etc/skel**

-s indica que o shell de traballo é **/bin/bash**

-d indica que a casa do usuario será **/home/user1**

-p indica que o contrasinal deste usuario é **abc123**.

\$ useradd -m -s /bin/bash -d /home/user2 -p \$(mkpasswd -m sha-256 abc123.) user2 #Crear o usuario de nome **user2**, onde:

-m crea o directorio **/home/user2** se este non existe, e ese directorio conterá o predefinido en **/etc/skel**

-s indica que o shell de traballo é **/bin/bash**

-d indica que a casa do usuario será **/home/user2**

-p indica que o contrasinal deste usuario é **abc123**.

\$ exit #Saír da consola de **root** e voltar á consola do usuario **user**

\$ su -c "touch /tmp/user1.txt" user1 #Como usuario **user1** creará o ficheiro **/tmp/user1.txt**. Para poder executar eses comandos como o usuario **user1** solicítase o contrasinal deste usuario.

Password:

\$ su -c 'touch /tmp/user2.txt' user2 #Como usuario **user2** creará o ficheiro **/tmp/user2.txt**. Para poder executar eses comandos como o usuario **user2** solicítase o contrasinal deste usuario.

Password:

\$ ls -ld /tmp && ls -l /tmp/user?.txt #Como usuario **user** listar de forma extendida o directorio **/tmp**, así como a listaxe extendida dos ficheiros **/tmp/user1.txt** e **/tmp/user2.txt**. O segundo comando executarase sempre e cando o primeiro teña éxito e non presente erro.

```
drwxrwxrwt 18 root root 15360 Abr 24 16:09 /tmp
-rw-r--r-- 1 user1 user1 0 Abr 24 16:05 /tmp/user1.txt
-rw-r--r-- 1 user2 user2 0 Abr 24 16:07 /tmp/user2.txt
```

\$ su -c 'rm -f /tmp/user2.txt' user1 #Como usuario **user1** intentar eliminar de forma forzosa o ficheiro **/tmp/user2.txt** pertencente ao usuario **user2**. Para poder executar ese comando como o usuario **user1** solicítase o contrasinal deste usuario.

Password:

rm: no se puede borrar '/tmp/user2.txt': Permiso denegado

Todo parece indicar que non se pode eliminar ese ficheiro xa que o usuario **user1**, non posúe permisos de escritura sobre ese ficheiro, xa que na máscara de permisos posúe os permisos **r--** correspondentes aos permisos **outros** dos permisos **ugo**

Entón imos dar permisos **ugo** de escritura a **outros** para que o usuario **user1** poida eliminar ese ficheiro:

\$ su -c 'chmod o+w /tmp/user2.txt' root #Como usuario **root** dar permisos de escritura a os **outros** nos permisos **ugo**. Para poder executar ese comando como o usuario **root** solicítase o contrasinal deste usuario.

Password:

\$ ls -ld /tmp && ls -l /tmp/user?.txt #Como usuario **user** listar de forma extendida o directorio **/tmp**, así como a listaxe extendida dos ficheiros **/tmp/user1.txt** e **/tmp/user2.txt**. O segundo comando executarase sempre e cando o primeiro teña éxito e non presente erro.

```
drwxrwxrwt 18 root root 15360 Abr 24 16:20 /tmp
-rw-r--r-- 1 user1 user1 0 Abr 24 16:05 /tmp/user1.txt
-rw-r--rw- 1 user2 user2 0 Abr 24 16:07 /tmp/user2.txt
```

\$ su -c 'rm -f /tmp/user2.txt' user1 #Voltamos como usuario **user1** a intentar eliminar de forma forzosa o ficheiro **/tmp/user2.txt** pertencente ao usuario **user2**. Para poder executar ese comando como o usuario **user1** solicítase o contrasinal deste usuario.

```
Password:
rm: no se puede borrar '/tmp/user2.txt': Permiso denegado
```

E agora, como é posible que non se elimine o ficheiro se o usuario **user1**, posúe permisos de escritura sobre ese ficheiro, xa que na máscara de permisos posúe os permisos **rw-** correspondentes aos permisos **outros** dos permisos **ugo**?. Ben, pois é debido ao permiso **Sticky (1000 ou S ou s)**, o cal indica, que no directorio onde está activo ese permiso soamente **root** ou o **usuario principal** dos permisos **ugo** dese directorio, ou os usuarios propietarios dos seus ficheiros(directorios) serán os que teñan permisos para renomear/eliminar eses ficheiros(directorios).

\$ su -c 'chgrp user1 /tmp/user2.txt' root #Como usuario **root** cambiar o grupo principal a **user1** nos permisos **ugo**. Para poder executar ese comando como o usuario **root** solicítase o contrasinal deste usuario.

\$ ls -ld /tmp && ls -l /tmp/user?.txt #Como usuario **user** listar de forma extendida o directorio **/tmp**, así como a listaxe extendida dos ficheiros **/tmp/user1.txt** e **/tmp/user2.txt**. O segundo comando executarase sempre e cando o primeiro teña éxito e non presente erro.

```
drwxrwxrwt 18 root root 15360 Abr 24 16:30 /tmp
-rw-r--r-- 1 user1 user1 0 Abr 24 16:05 /tmp/user1.txt
-rw-r--r-- 1 user2 user1 0 Abr 24 16:07 /tmp/user2.txt
```

\$ su -c 'chmod g+w /tmp/user2.txt' root #Como usuario **root** dar permisos de escritura ao grupo principal dos permisos **ugo**. Para poder executar ese comando como o usuario **root** solicítase o contrasinal deste usuario.

\$ ls -ld /tmp && ls -l /tmp/user?.txt #Como usuario **user** listar de forma extendida o directorio **/tmp**, así como a listaxe extendida dos ficheiros **/tmp/user1.txt** e **/tmp/user2.txt**. O segundo comando executarase sempre e cando o primeiro teña éxito e non presente erro.

```
drwxrwxrwt 18 root root 15360 Abr 24 16:30 /tmp
-rw-r--r-- 1 user1 user1 0 Abr 24 16:05 /tmp/user1.txt
-rw-rw-rw- 1 user2 user1 0 Abr 24 16:07 /tmp/user2.txt
```

\$ su -c 'rm -f /tmp/user2.txt' user1 #Como usuario **user1** intentar eliminar de forma forzosa o ficheiro **/tmp/user2.txt** pertencente ao usuario **user2**. Para poder executar ese comando como o usuario **user1** solicítase o contrasinal deste usuario.

```
Password:
rm: no se puede borrar '/tmp/user2.txt': Permiso denegado
```

E agora, como é posible que non se elimine o ficheiro se o usuario **user1**, posúe permisos de escritura sobre ese ficheiro, e o grupo principal é **user1** xa que na máscara de permisos posúe os permisos **rw-** correspondentes aos permisos **grupo principal** e **outros** dos permisos **ugo**?. Ben, pois é debido ao mesmo de antes, ao permiso **Sticky (1000 ou S ou s)**, o cal indica, que no directorio onde está activo ese permiso soamente **root** ou o **usuario principal** dos permisos **ugo** dese directorio, ou os usuarios propietarios dos seus ficheiros(directorios) serán os que teñan permisos para renomear/eliminar eses ficheiros(directorios).

\$ su -c 'chgrp user1 /tmp' root #Como usuario **root** cambiar ao directorio **/tmp** o grupo principal a **user1** nos permisos **ugo**. Para poder executar ese comando como o usuario **root** solicítase o contrasinal deste usuario.

\$ ls -ld /tmp && ls -l /tmp/user?.txt #Como usuario **user** listar de forma extendida o directorio **/tmp**, así como a listaxe extendida dos ficheiros **/tmp/user1.txt** e **/tmp/user2.txt**. O segundo comando executarase sempre e cando o primeiro teña éxito e non presente erro.

```
drwxrwxrwt 18 root user1 15360 Abr 24 16:39 /tmp
-rw-r--r-- 1 user1 user1 0 Abr 24 16:05 /tmp/user1.txt
-rw-rw-rw- 1 user2 user1 0 Abr 24 16:07 /tmp/user2.txt
```

\$ su -c 'rm -f /tmp/user2.txt' user1 #Como usuario **user1** intentar eliminar de forma forzosa o ficheiro **/tmp/user2.txt** pertencente ao usuario **user2**. Para poder executar ese comando como o usuario **user1** solicítase o contrasinal deste usuario.

```
Password:
rm: no se puede borrar '/tmp/user2.txt': Permiso denegado
```

E agora, como é posible que non se elimine o ficheiro se o usuario **user1**, posúe permisos de escritura sobre ese ficheiro, e o grupo principal é **user1** xa que na máscara de permisos posúe os permisos **rw-** correspondentes aos permisos **grupo principal** e **outros** dos permisos **ugo**? E ademais como é posible se agora o directorio **/tmp** ten agora como grupo principal a **user1**? Ben, pois é debido ao mesmo de antes, ao permiso **Sticky (1000 ou S ou s)**, o cal indica, que no directorio onde está activo ese permiso soamente **root** ou o **usuario principal** dos permisos **ugo** dese directorio, ou os usuarios propietarios dos seus ficheiros(directorios) serán os que teñan permisos para renomear/eliminar eses ficheiros(directorios).

\$ su -c 'chown user1 /tmp' root #Como usuario **root** cambiar ao directorio **/tmp** o usuario principal a **user1** nos permisos **ugo**. Para poder executar ese comando como o usuario **root** solicítase o contrasinal deste usuario.

\$ ls -ld /tmp && ls -l /tmp/user?.txt #Como usuario **user** listar de forma extendida o directorio **/tmp**, así como a listaxe extendida dos ficheiros **/tmp/user1.txt** e **/tmp/user2.txt**. O segundo comando executarase sempre e cando o primeiro teña éxito e non presente erro.

```
drwxrwxrwt 18 user1 user1 15360 Abr 24 16:47 /tmp
-rw-r--r-- 1 user1 user1 0 Abr 24 16:05 /tmp/user1.txt
-rw-rw-rw- 1 user2 user1 0 Abr 24 16:07 /tmp/user2.txt
```

\$ su -c 'rm -f /tmp/user2.txt' user1 #Como usuario **user1** intentar eliminar de forma forzosa o ficheiro **/tmp/user2.txt** pertencente ao usuario **user2**. Para poder executar ese comando como o usuario **user1** solicítase o contrasinal deste usuario.

```
Password:
```

E agora, si é posible eliminar o ficheiro xa que o usuario **user1**, é o usuario principal nos permisos **ugo** do directorio **/tmp** que posúe o **sticky bit** activado.

\$ ls -ld /tmp && ls -l /tmp/user?.txt #Como usuario **user** listar de forma extendida o directorio **/tmp**, así como a listaxe extendida dos ficheiros **/tmp/user1.txt** e **/tmp/user2.txt**. O segundo comando executarase sempre e cando o primeiro teña éxito e non presente erro.

```
drwxrwxrwt 18 user1 user1 15360 Abr 24 16:47 /tmp
-rw-r--r-- 1 user1 user1 0 Abr 24 16:05 /tmp/user1.txt
```

\$ su -c 'chown root. /tmp' root #Como usuario **root** cambiar ao directorio **/tmp** o usuario principal e grupo principal a **root** nos permisos **ugo**. Para poder executar ese comando como o usuario **root** solicítase o contrasinal deste usuario.

\$ ls -ld /tmp && ls -l /tmp/user?.txt #Como usuario **user** listar de forma extendida o directorio **/tmp**, así como a listaxe extendida dos ficheiros **/tmp/user1.txt** e **/tmp/user2.txt**. O segundo comando executarase sempre e cando o primeiro teña éxito e non presente erro.

```
drwxrwxrwt 18 root root 15360 Abr 24 17:00 /tmp
-rw-r--r-- 1 user1 user1 0 Abr 24 16:05 /tmp/user1.txt
```

Ricardo Feijoo Costa



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



# Kernel: Parámetros de arranque (II)



**ESCENARIO:** Para verificar o que acontece na práctica crear en VirtualBox unha máquina virtual coas seguintes características:

- 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.
    - Táboa de particións msdos
    - Dúas particións primarias:
      - Raíz do sistema: /dev/sda1 (/). Formato: ext4
      - Swap: /dev/sda2 (swap). Formato: swap
    - Nome de usuario: usuario
    - Nome computador: usuario-pc
    - Contraseña: abc123. (Olo que o contraseña ten un carácter punto final)
- Primeira opción de arranque: Óptica
- Segunda opción de arranque: Disco duro

## Práctica (init, runlevels, /etc/rcX.d/, /etc/init.d/)

1. Pór como primeira opción de arranque o disco duro.
2. Arrancar a máquina virtual en modo Inicio normal.
3. 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 ↑.
4. Seleccionar a primeira opción de arranque.
5. Premer a tecla **e** (edit) para poder editar os parámetros de arranque do kernel.
6. Moverse coa tecla frecha abaixo ↓ ata chegar á liña onde aparecen os parámetros **ro quiet splash**
7. 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). **Olo! Non está cargado completamente o sistema operativo, pero si está recoñecido o hardware.**
8. Executar:

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

# ip addr show #Amosar a configuración de rede das interfaces existentes na máquina virtual

# ping -c 4 www.google.es #Enviar 4 paquetes ICMP ECHO\_REQUEST a www.google.es, solicitando 4 paquetes ICMP ECHO\_RESPONSE, para verificar a conectividade de rede hacia Internet e ao servidor de google.

# ping -c 4 localhost #Enviar 4 paquetes ICMP ECHO\_REQUEST a www.google.es, solicitando 4 paquetes ICMP ECHO\_RESPONSE, para verificar a conectividade de rede hacia a interface local de loopback.

Como modificamos o proceso init de arranque, este non executou os scripts correspondentes ao nivel de arranque. Imos comprobar en que nivel de execución foi arrancado o sistema.

# runlevel #Buscar a información do nivel de arranque previo e actual.

Vemos que o runlevel é unknown, polo que, non se executaron scripts de arranque, é dicir, non se executaron os scripts /etc/rcX.d, sendo X o nivel de arranque, que apuntan aos scripts correspondentes en /etc/init.d/

# mount #Amosar os sistemas de ficheiros montados, é dicir, os que está a usar e podemos empregar no sistema operativo.

# cat /etc/fstab #Amosar o contido de /etc/fstab que contén a información estática sobre os sistemas de ficheiros, os cales son montados durante o arranque do sistema operativo.

# mount -a #Monta, todos os sistemas de ficheiros mencionados en /etc/fstab, según a orde na que aparecen, é dicir, de arriba a abaixo.

# mount #Amosar os sistemas de ficheiros montados, é dicir, os que está a usar e podemos empregar no sistema operativo.

Vemos que non existe diferencia entre a saída da execución de este comando con respecto a súa execución anterior. Polo tanto o problema de conectividade non depende aparentemente dos sistemas de ficheiros montados, é dicir, o hardware e comunicación coa interface debeu ser recoñecida. Imos ver entón se podemos configurar a tarxeta de rede mediante un servidor DHCP.

# dhclient -v enp0s3 #Solicitar a configuración de rede por DHCP para a interface enp0s3. A opción -v(verbose) amosa a saída do comando en modo detallado.

# ip addr show #Amosar a configuración de rede das interfaces existentes na máquina virtual

# ping -c 4 www.google.es #Agora obtemos conectividade ao enviar 4 paquetes ICMP ECHO\_REQUEST a www.google.es

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

# MBR (Primeiros 512B - Sector 0)

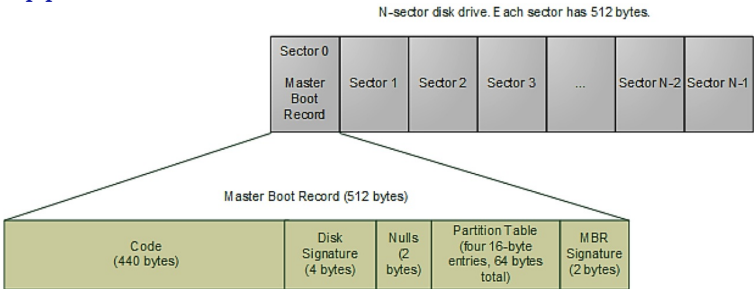
## Backup-Restore

NOTA: Para verificar o que acontece na prática criar unha máquina virtual en VirtualBox que arranque cunha ISO Live Debian 32bits, escritorio XFCE, 512MB de RAM e disco duro dinámico de 20GB. Imos supor que esta máquina virtual posúe o nome **Debian32-Instalacion** e o disco duro posúe o nome **Debian32-Instalacion.vdi**, o cal é arrancable e ten instalado o SO GNU/Linux Debian 9 de 32bits. Verificar que a primeira opción de arranque sexa o CD Virtual.

Algo moi importante e que todos deberiamos facer unha vez creadas as particións e/ou instalado/s o/s sistema/s operativo/s é facer unha copia de seguridade do MBR. O **backup pódese facer de varios xeitos**. Nesta práctica centrarémonos no comando **dd**.

Sector = Bloque

Sector lóxico != Sector físico



Flu-Project

### Práctica Backup/Restore MBR mediante o comando dd

1. Arrancar a máquina virtual creada en modo Inicio normal

`VBoxManage startvm Debian32-Instalacion`

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

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

`$ sudo su -` #Acceder á consola de root(administrador) a través dos permisos configurados co comando sudo (/etc/sudoers, visudo)

#### Backup MBR

`# dd if=/dev/sda of=/tmp/backup-mbr.img bs=512 count=1` #Facer unha copia de seguridade do MBR do disco /dev/sda e volcalo no ficheiro /tmp/backup-mbr.img. O argumento bs=512 indica o número de bytes a copiar. O argumento count=1 indica a cantidade de bloques(sectores) a copiar.

Enviar esa copia a un pendrive, disco duro, rede(**ssh**), e-mail, etc.

#### Eliminar o MBR

`# dd if=/dev/zero of=/dev/sda bs=512 count=1` #Sobreescribir o MBR do disco /dev/sda con ceros. O argumento bs=512 indica o número de bytes a copiar. O argumento count=1 indica a cantidade de bloques(sectores) a copiar.

`# init 0` #Apagar o sistema.

3. Desconectar o cd virtual (ISO) da máquina virtual.

4. Iniciar a máquina virtual e comprobar que agora NON arranca o sistema operativo.

`VBoxManage startvm Debian32-Instalacion`

5. Voltar a conectar a ISO á máquina virtual

6. Arrancar a máquina virtual en modo Inicio normal

`VBoxManage startvm Debian32-Instalacion`

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

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

`$ sudo su -` #Acceder á consola de root(administrador) a través dos permisos configurados co comando sudo (/etc/sudoers, visudo)

#### Restore MBR

Recuperar o arquivo copia backup-mbr.img do pendrive, disco duro, rede(**ssh**), e-mail, etc. e copialo en /tmp.

`# dd if=/tmp/backup-mbr.img of=/dev/sda bs=512 count=1` #Restaurar unha copia de seguridade do MBR do disco /dev/sda e volcala no disco. O argumento bs=512 indica o número de bytes a copiar. O argumento count=1 indica a cantidade de bloques(sectores) a copiar.

`# init 0` #Apagar o sistema.

8. Desconectar o cd virtual (ISO) da máquina virtual.

9. Iniciar a máquina virtual e comprobar que agora SI arranca o sistema operativo.

`VBoxManage startvm Debian32-Instalacion`

Ricardo Feijoo Costa



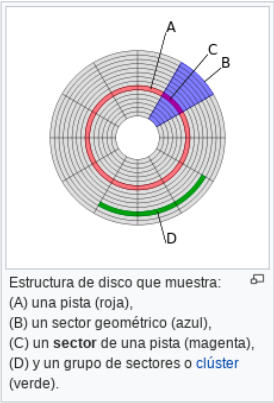
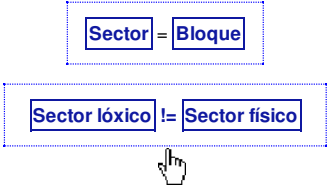
This work is licensed under a **Creative Commons Attribution-ShareAlike 4.0 International License**

# Sistemas de arquivos: Inodos

## Borrado de ficheiros non implica perda de información

NOTA: Para verificar o que acontece na práctica crear unha máquina virtual en VirtualBox que arranque cunha ISO Live Debian 32bits, escritorio XFCE, 512MB de RAM e disco duro dinámico de 8GB. Imos supor que esta máquina virtual posúe o nome **Debian32-Recovery** e o disco duro posúe o nome **Debian32-Recovery.vdi**. Verificar que a primeira opción de arranque sexa o CD Virtual.

Por cada ficheiro ou directorio no sistema, existe un **inodo**, unha estrutura de datos, que garda a información do ficheiro. É similar aos rexistros do MFT en NTFS.



Wikipedia



### Comandos de interese sobre sistemas de ficheiros ext2/ext3/ext4

- **debugfs** #O comando debugfs permite depurar sistemas de ficheiros ext2/ext3/ext4
- **stat** #O comando stat permite amosar información sobre ficheiros ou sistemas de ficheiros.
- **dumpe2fs** #O comando dumpe2fs permite listar información sobre sistemas de ficheiros ext2/ext3/ext4.
- **tune2fs** #O comando tune2fs permite axustar os parámetros do sistema de ficheiros sobre sistemas de ficheiros ext2/ext3/ext4.

### Práctica Borrado de ficheiros non implica perda de información

1. Arrancar a máquina virtual creada en modo Inicio normal

**VBoxManage startvm Debian32-Recovery**

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

```
$ setxkbmap es #Cambiar o mapa de teclado ao idioma español.  
$ sudo su - #Acceder á consola de root(administrador) a través dos permisos configurados co comando sudo (/etc/sudoers, visudo)
```

#### Crear e formatear particións

```
# parted --script /dev/sda mklabel msdos #Crear a etiqueta de disco (táboa de particións) ao dispositivo /dev/sda sen ter que acceder ao prompt de parted  
  
# parted --script /dev/sda mkpart primary 0 50% -a cylinder #Crear unha partición primaria no disco /dev/sda cos primeiros 5GB, alineando a cilindros, sen ter que acceder ao prompt de parted
```

```
# parted --script /dev/sda mkpart primary 50% 70% -a cylinder #Crear unha partición primaria no disco /dev/sda de 2GB a continuación da partición de 5GB, alineando a cilindros, sen ter que acceder ao prompt de parted
```

```
# parted --script /dev/sda print #Amosa a táboa de particións do disco /dev/sda
```

```
Model: ATA VBOX HARDDISK (scsi)  
Disk /dev/sda: 8590MB  
Sector size (logical/physical): 512B/512B  
Partition Table: msdos  
Disk Flags:
```

Number	Start	End	Size	Type	File system	Flags
1	32.3kB	4294MB	4294MB	primary		
2	4294MB	6013MB	1719MB	primary		

```
# mkfs.ext4 -F -L 'PARTICION1' /dev/sda1 #Formatear en ext4 coa etiqueta PARTICION1 a partición primaria /dev/sda1
```

```
mke2fs 1.43.4 (31-Jan-2017)  
/dev/sda1 contains a ext4 file system labelled 'PARTICION1'  
last mounted on /mnt/recuperacion on Mon Oct 29 17:58:50 2018  
Creating filesystem with 1048233 4k blocks and 262144 inodes  
Filesystem UUID: 459fb916-7189-4b3f-83de-fd81b56973f8  
Superblock backups stored on blocks:  
32768, 98304, 163840, 229376, 294912, 819200, 884736
```

```
Allocating group tables: done  
Writing inode tables: done  
Creating journal (16384 blocks): done  
Writing superblocks and filesystem accounting information: done
```

```
# mkfs.ext3 -F -L 'PARTICION2' /dev/sda2 #Formatear en ext3 coa etiqueta PARTICION2 a partición primaria /dev/sda2
```

```
mke2fs 1.43.4 (31-Jan-2017)  
/dev/sda2 contains a ext3 file system labelled 'PARTICION2'  
created on Mon Oct 29 17:58:29 2018  
Creating filesystem with 419698 4k blocks and 105040 inodes  
Filesystem UUID: d5c70817-fa3c-4734-8825-fa78463558cd  
Superblock backups stored on blocks:  
32768, 98304, 163840, 229376, 294912
```

```
Allocating group tables: done  
Writing inode tables: done  
Creating journal (8192 blocks): done  
Writing superblocks and filesystem accounting information: done
```

### Montar particións e crear ficheiros e directorios

```
# mkdir /mnt/recuperacion #Crear o directorio /mnt/recuperacion

# mount /dev/sda1 /mnt/recuperacion #Montar (facer uso) a partición primaria /dev/sda1 en /mnt/recuperacion

# cd /mnt/recuperacion #Acceder ao directorio /mnt/recuperacion

# mkdir proverbios #Crear o directorio /mnt/recuperacion/proverbios:

# cd proverbios #Acceder ao directorio /mnt/recuperacion/proverbios:

# echo 'Aprender sen pensar é inútil. Pensar sen aprender, perigoso. Confucio' > Confucio1.txt #Crear o ficheiro /mnt/recuperacion/proverbios/Confucio1.txt co contido 1 frase.

# echo 'Eu non procuro coñecer as preguntas; procuro coñecer as respostas. Confucio' > Confucio2.txt #Crear o ficheiro /mnt/recuperacion/proverbios/Confucio2.txt co contido 1 frase.

# echo 'Estudia o pasado se queres pronosticar o futuro. Confucio' > Confucio3.txt #Crear o ficheiro /mnt/recuperacion/proverbios/Confucio3.txt co contido 1 frase.

# ls -lia #Listar de forma extendida e amosar os inodos dos ficheiros e directorios contidos en /mnt/recuperacion/proverbios

total 20
131073 drwxr-xr-x 2 root root 4096 Nov 1 00:25 .
2 drwxr-xr-x 4 root root 4096 Nov 1 00:24 ..
131074 -rw-r--r-- 1 root root 72 Nov 1 00:24 Confucio1.txt
131075 -rw-r--r-- 1 root root 69 Nov 1 00:25 Confucio2.txt
131076 -rw-r--r-- 1 root root 58 Nov 1 00:25 Confucio3.txt

# cat Confucio1.txt Confucio2.txt Confucio3.txt #Ver os contidos dos ficheiros Confucio1.txt Confucio2.txt Confucio3.txt

Aprender sen pensar é inútil. Pensar sen aprender, perigoso. Confucio
Eu non procuro coñecer as preguntas; procuro coñecer as respostas. Confucio
Estudia o pasado se queres pronosticar o futuro. Confucio
```

### Desmontar para poder intentar recuperar a información

```
# cd #Acceder ao directorio casa do usuario (/home/user)

# umount /mnt/recuperacion #Desmontar (deixar de facer uso) a partición primaria /dev/sda1 que estaba montada en /mnt/recuperacion
```

### Revisar os bloques que referencian a información.

```
# debugfs /dev/sda1 #Executar o comando debugfs sobre a partición primaria /dev/sda1

debugfs 1.43.4 (31-Jan-2017)
debugfs: stat <131074> #Ver información sobre o inodo 131074
Inode: 131074 Type: regular Mode: 0644 Flags: 0x80000
Generation: 204948144 Version: 0x00000000:00000001
User:0 Group:0 Project:0 Size: 72
File ACL: 0 Directory ACL: 0
Links: 1 Blockcount: 8
Fragment: Address: 0 Number: 0 Size: 0
ctime: 0x5bda47db:8ec2f228 -- Thu Nov 1 00:24:59 2018
atime: 0x5bda4810:9270b1e4 -- Thu Nov 1 00:25:52 2018
mtime: 0x5bda47db:8ec2f228 -- Thu Nov 1 00:24:59 2018
crtime: 0x5bda47db:8ec2f228 -- Thu Nov 1 00:24:59 2018
Size of extra inode fields: 32
Inode checksum: 0x53cfeeb
EXTENTS:
(0):557056
(END)

Vemos que logo de EXTENTS aparece/n o/s bloque/s onde está gardada a información. Premer a tecla q para voltar á consola debugfs

debugfs: cat <131074> #Ver o contido do ficheiro que corresponde co inodo 131074.
Aprender sen pensar é inútil. Pensar sen aprender, perigoso. Confucio
debugfs: blocks <131074> #Ver o/s bloque/s que apunta/n ao contido do ficheiro que corresponde co inodo 131074
557056

debugfs: q #Premer de novo a tecla q para saír da consola debugfs
```

### Eliminar ficheiros

```
# mount /dev/sda1 /mnt/recuperacion #Montar (facer uso) a partición primaria /dev/sda1 en /mnt/recuperacion

# cd /mnt/recuperacion/proverbios #Acceder ao directorio /mnt/recuperacion/proverbios

# rm Confucio1.txt #Borrar o ficheiro /mnt/recuperacion/proverbios/Confucio1.txt
```

### Desmontar para revisar o que pasou coa información

```
# cd #Acceder ao directorio casa do usuario (/home/user)
```

```
# umount /mnt/recuperacion #Desmontar (deixar de facer uso) a partición primaria /dev/sda1 que estaba montada en /mnt/recuperacion
```

### Revisar os bloques que referencian a información.

```
# debugfs /dev/sda1 #Executar o comando debugfs sobre a partición primaria /dev/sda1
```

```
debugfs 1.43.4 (31-Jan-2017)
debugfs: stat <131074> #Ver información sobre o inodo 131074
Inode: 131074 Type: regular Mode: 0644 Flags: 0x80000
Generation: 204948144 Version: 0x00000000:00000001
User:0 Group:0 Project:0 Size: 0
File ACL: 0 Directory ACL: 0
Links: 0 Blockcount: 0
Fragment: Address: 0 Number: 0 Size: 0
ctime: 0x5bda4b93:21e4db74 -- Thu Nov 1 00:40:51 2018
atime: 0x5bda4810:9270b1e4 -- Thu Nov 1 00:25:52 2018
mtime: 0x5bda4b93:21e4db74 -- Thu Nov 1 00:40:51 2018
crtime: 0x5bda47db:8ec2f228 -- Thu Nov 1 00:24:59 2018
dtime: 0x5bda4b93:(21e4db74) -- Thu Nov 1 00:40:51 2018
Size of extra inode fields: 32
Inode checksum: 0xcd1b5c5b
EXTENTS
(END)
```

Vemos que logo de EXTENTS non aparece ningún bloque onde está gardada a información, é dicir, ao borrar o arquivo perdeuse a referencia dos bloques correspondnetes ao contido pero a información segue existindo a non ser que fose sobreescrita. Premer a tecla **q** para voltar á consola debugfs

```
debugfs: cat <131074> #Ver o contido do ficheiro que corresponde co inodo 131074.
```

Neste caso non vemos contido xa que o ficheiro foi eliminado.

```
debugfs: blocks <131074> #Ver o/s bloque/s que apunta/n ao contido do ficheiro que corresponde co inodo 131074
```

Neste caso non vemos referencia ningún bloque xa que o ficheiro foi eliminado perdendo así a referencia ao/s bloque/s. Como anteriormente vimos o/s bloque/s referenciados (**557056**) ao contido do ficheiro Confucio1.txt imos revisar se podemos ver o contido dese bloque. E no caso de poder ver o contido como non sobreescibimos o/s bloque/s deberiamos ver o texto do ficheiro Confucio1.txt

```
debugfs: block_dump 557056
0000 4170 7265 6e64 6572 2073 656e 2070 656e Aprender sen pen
0020 7361 7220 c3a9 2069 6ec3 ba74 696c 2e20 sar .. in..til.
0040 5065 6e73 6172 2073 656e 2061 7072 656e Pensar sen apren
0060 6465 722c 2070 6572 6967 6f73 6f2e 2043 der, perigoso. C
0100 6f6e 6675 6369 6f0a 0000 0000 0000 0000 onfucio.....
0120 0000 0000 0000 0000 0000 0000 0000 .....
*
```

Acabamos de ver que o bloque non está referenciado a ningún ficheiro pero segue preservando o contido do ficheiro Confucio1.txt porque non foi reescrito por ningún outro arquivo que o referencie.

#### (dd): Recuperar a información do ficheiro borrado

Imos revisar o tamaño de bloque da partición /dev/sda1

# dumpe2fs -h /dev/sda1 | grep -i 'block size' #Listar o tamaño de bloque en bytes mediante o comando dumpe2fs. O comando dumpe2fs permite listar información sobre sistemas de ficheiros ext2/ext3/ext4. A opción -h permite ver o tamaño do bloque, é dicir, do contido de información que ofrece o superbloque permite ver o tamaño do bloque. O comando grep -i permite polo patrón de búsqueda, neste caso 'block size', ignorando a diferenza entre maiúsculas e minúsculas.

Block size: 4096

# tune2fs -l /dev/sda1 | grep -i 'block size' #Listar o tamaño de bloque en bytes mediante o comando tune2fs. O comando tune2fs permite axustar os parámetros do sistema de ficheiros sobre sistemas de ficheiros ext2/ext3/ext4. A opción -l permite ver os contidos do superbloque do sistema de ficheiros, é dicir, o contido de información sobre o sistema de ficheiros que ofrece o superbloque. O comando grep -i permite polo patrón de búsqueda, neste caso 'block size', ignorando a diferenza entre maiúsculas e minúsculas.

Block size: 4096

# stat -fc %s . #Listar o tamaño de bloque mediante o comando stat. O comando stat permite amosar información sobre ficheiros ou sistemas de ficheiros. A opción -f permite amosar información sobre o sistema de ficheiros e non sobre ficheiros. A opción -c permite formatear a saída a ensinar. O argumento %s amosa o tamaño total en bytes

4096

Todos os comandos anteriores amosan que o tamaño do bloque en disco é de: **4096B** ou **4.0kB**

# dd if=/dev/sda1 of=recovery-data.txt bs=4096 count=1 skip=557056 #Recuperación do bloque 557056 mediante o comando dd. Recuperamos da partición /dev/sda1 no ficheiro recovery-data.txt o ficheiro eliminado Confucio1.txt. Para iso ao comando dd pasámolle como argumentos o valor do bloque (bs=4096), a cantidade de bloques a recuperar (count=1) e dende que bloque comezar a recuperar (skip=557056).

1+0 records in  
1+0 records out  
4096 bytes (4.1 kB, 4.0 KiB) copied, 0.00274002 s, 1.5 MB/s

# cat recovery-data.txt #Ver o contido do ficheiro recovery-data.txt

Aprender sen pensar é inútil. Pensar sen aprender, perigoso. Confucio

**O contido do ficheiro foi recuperado**

Ricardo Feijoo Costa



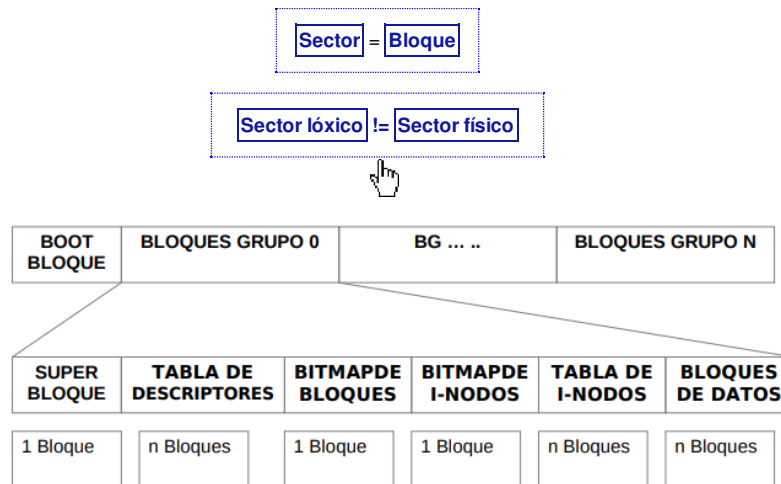
This work is licensed under a [Creative Commons Attribution-ShareAlike 4.0 International License](https://creativecommons.org/licenses/by-sa/4.0/)

# Sistemas de arquivos: Superbloque

## Información sobre o sistema de ficheiros

**ESCENARIO:** Para verificar o que acontece na práctica crear en VirtualBox 1 máquina virtual coas seguintes características:

- Nome: **Debian32-Instalacion**
- ISO Live Debian 32bits, escritorio XFCE, 2 discos duros dinámicos.
- Primer disco duro:
  - Nome: **Debian32-Instalacion.vdi**
  - Arrancable
  - Ten instalado o SO GNU/Linux Debian 9 de 32bits.
- Segundo disco duro:
  - Nome: **Backup.vdi**
  - Tamaño: 8GB.
  - Soamente unha partición primaria.
  - Formato: FAT32.
- Primeira opción de arranque: Disco duro Debian32-Instalacion.
- No xestor de arranque escoller a primeira opción (opción por defecto).



Diseño de Sistemas Operativos. Universidad de Las Palmas de Gran Canaria

O **Superbloque** é un bloque que contén a información máis relevante e describe o sistema de ficheiros.

Cada grupo de bloques contén unha copia do superbloque.

O superbloque contén 3 tipos de parámetros:

1. **Parámetros non modificables:** Creados cando se xera o sistema de ficheiros.
  - Tamaño de bloque
  - Número de bloques dispoñibles
  - Número de inodos dispoñibles
2. **Parámetros modificables** durante o uso do sistema de ficheiros:
  - Bloques reservados ao superusuario
  - UID (identificador usuario) do superusuario por defecto
  - GID (identificador de grupo) do superusuario por defecto
3. **Parámetros de estado e uso** que varían durante o uso do sistema de ficheiros:
  - Inodos e bloques libres
  - Estado do sistemas de ficheiros (limpo/non limpo)
  - Número de montaxes
  - Máximo número de montaxes entre comprobacións
  - Máximo tempo entre comprobacións
  - Comportamento fronte a erros
  - Directorio da última montaxe

### Comandos de interese sobre sistemas de ficheiros ext2/ext3/ext4

- **fdisk** #O comando fdisk permite manipular as táboas de particións para Linux.
- **dumpe2fs** #O comando dumpe2fs permite listar información sobre sistemas de ficheiros ext2/ext3/ext4. Amosa información sobre o superbloque e grupos de bloques.
- **mke2fs** #O comando mke2fs permite crear sistemas de ficheiros ext2/ext3/ext4. Coa opción -n non xera sistemas de ficheiros pero permite amosar a localización dos bloques de backup do superbloque.
- **dd** #O comando dd converte e copia un ficheiro. Copia un ficheiro cun tamaño de bloque seleccionable polo usuario, ao mesmo tempo que, opcionalmente, realiza sobre el certas conversións.
- **fsck** #O comando fsck permite chequear e reparar un sistemas de ficheiros Linux.
- **e2fsck** #O comando e2fsck permite chequear e reparar un sistema de ficheiros Linux ext2/ext3/ext4.

### Amosar información sobre os superbloques e grupos de bloques

# fdisk -l /dev/sda #Lista a táboa de particións do disco /dev/sda e logo remata.

# dumpe2fs /dev/sda1 #Amosar información sobre o superbloque e grupos de bloques da partición /dev/sda1

# mke2fs -n /dev/sda1 #Amosar a lista dos superbloques, é dicir cales son os bloques que gardan a información do superbloque en /dev/sda1

## Práctica

1. Arrancar a máquina virtual creada en modo Inicio normal

**VBoxManage startvm Debian32-Instalacion**

2. Comprobar que arranca o sistema operativo Debian 9.

3. Apagar a máquina virtual.

**VBoxManage controlvm Debian32-Instalacion poweroff**

4. Modificar na configuración da máquina virtual as opcións de arranque e pór como primeira opción o CD Virtual.

5. Arrancar de novo a máquina virtual creada en modo Inicio normal

**VBoxManage startvm Debian32-Instalacion**

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

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

\$ sudo su - #Acceder á consola de root(administrador) a través dos permisos configurados co comando sudo (/etc/sudoers, visudo)

### Amosar información sobre os superbloques

# fdisk -l /dev/sda #Lista a táboa de particións do disco /dev/sda

```
fdisk /dev/sda: 20 GiB, 21474836480 bytes, 41943040 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
Disk identifier: 0x89234789
```

Device	Boot	Start	End	Sectors	Size	Id	Type
/dev/sda1	*	2048	8814591	8812544	4.2G	83	Linux
/dev/sda2		8816638	41940991	33124354	15.8G	5	Extended
/dev/sda5		8816640	12402687	3586048	1.7G	83	Linux
/dev/sda6		12404736	13449215	1044480	510M	82	Linux swap / Solaris
/dev/sda7		13451264	14221311	770048	376M	83	Linux
/dev/sda8		14223360	41940991	27717632	13.2G	83	Linux

# dumpe2fs -h /dev/sda1 #Amosar información soamente sobre o superbloque e non os grupos de bloques da partición /dev/sda1

```
dumpe2fs 1.43.4 (31-Jan-2017)
Filesystem volume name:
Last mounted on: /
Filesystem UUID: b85b3fe0-3c31-4dc6-92d6-83477c758016
Filesystem magic number: 0xEF53
Filesystem revision #: 1 (dynamic)
Filesystem features: has_journal ext_attr resize_inode dir_index filetype extent 64bit flex_bg sparse_super large_file huge_file dir_nlink extra_isize metadata_csum
Filesystem flags: signed_directory_hash
Default mount options: user_xattr acl
Filesystem state: clean
Errors behavior: Continue
Filesystem OS type: Linux
Inode count: 275808
Block count: 1101568
Reserved block count: 55078
Free blocks: 378276
Free inodes: 179788
First block: 0
Block size: 4096
Fragment size: 4096
Group descriptor size: 64
Reserved GDT blocks: 537
Blocks per group: 32768
Fragments per group: 32768
Inodes per group: 8112
Inode blocks per group: 507
Flex block group size: 16
Filesystem created: Tue Nov 6 20:58:01 2018
Last mount time: Tue Nov 6 21:19:41 2018
Last write time: Tue Nov 6 21:19:39 2018
Mount count: 2
Maximum mount count: -1
Last checked: Tue Nov 6 20:58:01 2018
Check interval: 0 ()
Lifetime writes: 3452 MB
Reserved blocks uid: 0 (user root)
Reserved blocks gid: 0 (group root)
First inode: 11
Inode size: 256
Required extra isize: 32
Desired extra isize: 32
Journal inode: 8
Default directory hash: half_md4
Directory Hash Seed: fc2062ad-2d8b-4bc7-8ef9-c0f5bf613ba8
Journal backup: inode blocks
Checksum type: crc32c
Checksum: 0x7def9d51
Journal features: journal_incompat_revoke journal_64bit journal_checksum_v3
Journal size: 64M
Journal length: 16384
Journal sequence: 0x00000a0f
Journal start: 0
Journal checksum type: crc32c
Journal checksum: 0x53fcd0c0
```



# duple2fs -h /dev/sda1 | grep -i 'block size' #Listar o tamaño do bloque en bytes mediante o comando duple2fs. O comando duple2fs permite listar información sobre sistemas de ficheiros ext2/ext3/ext4. A opción -h permite ver o tamaño do bloque, é dicir, do contido de información que ofrece o superbloque permite ver o tamaño do bloque. O comando grep -i permite polo patrón de búsqueda, neste caso 'block size', ignorando a diferenza entre maiúsculas e minúsculas.

```
duple2fs 1.43.4 (31-Jan-2017)
Block size:      4096
```

# mke2fs -n /dev/sda1 #Amosar a lista dos superbloques, é dicir cales son os bloques que gardan a información do superbloque en /dev/sda1

```
mke2fs 1.43.4 (31-Jan-2017)
/dev/sda1 contains a ext4 file system
last mounted on / on Tue Nov  6 21:19:41 2018
Proceed anyway? (y,N) y
Creating filesystem with 1101568 4k blocks and 275808 inodes
Filesystem UUID: 1af8a123-2401-42a9-8393-379e534a7b75
Superblock backups stored on blocks:
32768, 98304, 163840, 229376, 294912, 819200, 884736
```

### Backup do superbloque.

# mkdir /media/recuperacion #Crear o directorio /media/recuperacion

# mount /dev/sdb1 /media/recuperacion #Montar (facer uso) a partición primaria /dev/sdb1 en /media/recuperacion

# dd if=/dev/sda1 of=/media/recuperacion/backup-superbloque.img bs=4096 count=1 skip=32768 #Facer unha copia do superbloque (bloque 32768) no arquivo /media/recuperacion/backup-superbloque.img. Para iso ao comando dd pásamolle como argumentos o tamaño do bloque en Bytes (bs=4096), a cantidade de bloques a recuperar (count=1) e dende que bloque comezar a recuperar (skip=32768).

# umount /media/recuperacion #Desmontar (deixar de facer uso) a partición primaria /dev/sdb1 que estaba montada en /media/recuperacion

### Eliminar o contido do superbloque.

# dd if=/dev/zero of=/dev/sda1 bs=4096 count=1 skip=32768 #Sobreescribir o bloque 32768 da partición /dev/sda1 con ceros. O argumento bs=4096 indica o número de bytes a copiar. O argumento count=1 indica a cantidde de bloques(sectores) a copiar. O argumento skip=32768 indica en que bloque comezar a copiar.

### Comprobar o arranque do sistema operativo.

7. Apagar a máquina virtual.

**VBoxManage controlvm Debian32-Instalacion poweroff**

8. Modificar na configuración da máquina virtual as opcións de arranque e pór como primeira opción o Disco duro Virtual.

9. Arrancar de novo a máquina virtual creada en modo Inicio normal

**VBoxManage startvm Debian32-Instalacion**

10. Comprobar o arranque do sistema operativo.

```
error: unknown filesystem.
Entering rescue mode...
grub rescue> _
```

11. Apagar a máquina virtual.

**VBoxManage controlvm Debian32-Instalacion poweroff**

12. Modificar na configuración da máquina virtual as opcións de arranque e pór como primeira opción o CD Virtual.

13. Arrancar de novo a máquina virtual creada en modo Inicio normal

**VBoxManage startvm Debian32-Instalacion**

### Recuperar o contido do superbloque.

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

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

\$ sudo su - #Acceder á consola de root(administrador) a través dos permisos configurados co comando sudo (/etc/sudoers, visudo)

# mkdir /media/recuperacion #Crear o directorio /media/recuperacion

# mount /dev/sdb1 /media/recuperacion #Montar (facer uso) a partición primaria /dev/sdb1 en /media/recuperacion

# dd if=/media/recuperacion/backup-superbloque.img of=/dev/sda1 bs=4096 count=1 skip=32768 #Recuperar a copia do superbloque (bloque 32768) existente no arquivo /media/recuperacion/backup-superbloque.img. Para iso ao comando dd pásamolle como argumentos o tamaño do bloque en Bytes (bs=4096), a cantidade de bloques a recuperar (count=1) e dende que bloque comezar a recuperar (skip=32768).

```
dd: /media/recuperacion/backup-superbloque.img: cannot skip to specified offset
0+0 records in
0+0 records out
0 bytes copied, 0.00560906 s, 0.0 kB/s
```

# fsck /dev/sda1 #Facer un chequeo da partición /dev/sda1

```
fsck /dev/sda1
fsck from util-linux 2.29.2
e2fsck 1.43.4 (31-Jan-2017)
ext2fs_open2: Bad magic number in super-block
fsck.ext4: Superblock invalid, trying backup blocks...
/dev/sda1 was not cleanly unmounted, check forced.
```

Pass 1: Checking inodes, blocks, and sizes  
 Pass 2: Checking directory structure  
 Pass 3: Checking directory connectivity  
 Pass 4: Checking reference counts  
 Pass 5: Checking group summary information  
 Block bitmap differences: +(884736-885274)  
 Fix? yes  
 Free blocks count wrong for group #0 (24079, counted=20599).  
 Fix? yes  
 Free blocks count wrong for group #1 (32229, counted=4354).  
 Fix? yes  
 Free blocks count wrong for group #2 (32768, counted=2533).  
 Fix? yes  
 Free blocks count wrong for group #3 (32229, counted=9995).  
 Fix? yes  
 Free blocks count wrong for group #4 (32768, counted=5405).  
 Fix? yes  
 Free blocks count wrong for group #5 (32229, counted=7333).  
 Fix? yes  
 Free blocks count wrong for group #6 (32768, counted=5606).  
 Fix? yes  
 Free blocks count wrong for group #7 (32229, counted=6600).  
 Fix? yes  
 Free blocks count wrong for group #8 (32768, counted=2965).  
 Fix? yes  
 Free blocks count wrong for group #9 (32229, counted=2965).  
 Fix? yes  
 Free blocks count wrong for group #10 (32768, counted=5881).  
 Fix? yes  
 Free blocks count wrong for group #11 (32768, counted=3126).  
 Fix? yes  
 Free blocks count wrong for group #12 (32768, counted=3964).  
 Fix? yes  
 Free blocks count wrong for group #13 (32768, counted=4745).  
 Fix? yes  
 Free blocks count wrong for group #14 (32768, counted=4077).  
 Fix? yes  
 Free blocks count wrong for group #15 (16384, counted=0).  
 Fix? yes  
 Free blocks count wrong for group #16 (24624, counted=19419).  
 Fix? yes  
 Free blocks count wrong for group #17 (32768, counted=2).  
 Fix? yes  
 Free blocks count wrong for group #18 (32768, counted=0).  
 Fix? yes  
 Free blocks count wrong for group #19 (32768, counted=0).  
 Fix? yes  
 Free blocks count wrong for group #20 (32768, counted=3665).  
 Fix? yes  
 Free blocks count wrong for group #21 (32768, counted=1549).  
 Fix? yes  
 Free blocks count wrong for group #22 (32768, counted=2784).  
 Fix? yes  
 Free blocks count wrong for group #23 (32768, counted=3529).  
 Fix? yes  
 Free blocks count wrong for group #24 (32768, counted=5204).  
 Fix? yes  
 Free blocks count wrong for group #25 (32229, counted=4288).  
 Fix? yes  
 Free blocks count wrong for group #26 (32768, counted=32413).  
 Fix? yes  
 Free blocks count wrong (1063560, counted=378276).  
 Fix? yes  
 Free inodes count wrong for group #0 (8101, counted=7).  
 Fix? yes  
 Directories count wrong for group #0 (2, counted=1496).  
 Fix? yes  
 Free inodes count wrong for group #1 (8112, counted=0).  
 Fix? yes  
 Directories count wrong for group #1 (0, counted=287).  
 Fix? yes  
 Free inodes count wrong for group #2 (8112, counted=0).  
 Fix? yes  
 Directories count wrong for group #2 (0, counted=578).  
 Fix? yes  
 Free inodes count wrong for group #3 (8112, counted=0).  
 Fix? yes  
 Directories count wrong for group #3 (0, counted=492).  
 Fix? yes  
 Free inodes count wrong for group #4 (8112, counted=6376).  
 Fix? yes  
 Directories count wrong for group #4 (0, counted=287).  
 Fix? yes  
 Free inodes count wrong for group #16 (8112, counted=0).  
 Fix? yes  
 Directories count wrong for group #16 (0, counted=961).  
 Fix? yes  
 Free inodes count wrong for group #17 (8112, counted=0).  
 Fix? yes  
 Directories count wrong for group #17 (0, counted=538).  
 Fix? yes  
 Free inodes count wrong for group #18 (8112, counted=0).  
 Fix? yes  
 Free inodes count wrong for group #19 (8112, counted=0).  
 Fix? yes  
 Directories count wrong for group #19 (0, counted=437).  
 Fix? yes  
 Free inodes count wrong for group #20 (8112, counted=0).  
 Fix? yes  
 Directories count wrong for group #20 (0, counted=772).

```

Fix? yes
Free inodes count wrong for group #21 (8112, counted=0).
Fix? yes
Directories count wrong for group #21 (0, counted=554).
Fix? yes
Free inodes count wrong for group #22 (8112, counted=0).
Fix? yes
Directories count wrong for group #22 (0, counted=522).
Fix? yes
Free inodes count wrong for group #23 (8112, counted=3231).
Fix? yes
Directories count wrong for group #23 (0, counted=498).
Fix? yes
Free inodes count wrong for group #24 (8112, counted=7934).
Fix? yes
Directories count wrong for group #24 (0, counted=178).
Fix? yes
Free inodes count wrong (275797, counted=179788).
Fix? yes
Inode bitmap differences: Group 0 inode bitmap does not match checksum.
FIXED.
/dev/sda1: ***** FILE SYSTEM WAS MODIFIED *****
/dev/sda1: 96020/275808 files (0.1% non-contiguous), 723292/1101568 blocks

```

### En vez do comando anterior poderíamos empregar este outro comando:

# e2fsck -b 32768 /dev/sda1 #Esta opción é empregada normalmente cando o superbloque primario está corrupto. A localización do superbloque de backup depende do tamaño de bloque do sistema de ficheiros. Así, se o tamaño do bloque é 1K o superbloque de backup atópase en 8193; se o tamaño é 2K atópase en 16384; e para 4K atópase en 32768

```

e2fsck 1.43.4 (31-Jan-2017)
/dev/sda1 was not cleanly unmounted, check forced.
Pass 1: Checking inodes, blocks, and sizes
Pass 2: Checking directory structure
Pass 3: Checking directory connectivity
Pass 4: Checking reference counts
Pass 5: Checking group summary information
Block bitmap differences: +(884736--885274)
Fix? yes
Free blocks count wrong for group #0 (24079, counted=20599).
Fix? yes
Free blocks count wrong for group #1 (32229, counted=4354).
Fix? yes
Free blocks count wrong for group #2 (32768, counted=2533).
Fix? yes
Free blocks count wrong for group #3 (32229, counted=9995).
Fix? yes
Free blocks count wrong for group #4 (32768, counted=5405).
Fix? yes
Free blocks count wrong for group #5 (32229, counted=7333).
Fix? yes
Free blocks count wrong for group #6 (32768, counted=5606).
Fix? yes
Free blocks count wrong for group #7 (32229, counted=6600).
Fix? yes
Free blocks count wrong for group #8 (32768, counted=2965).
Fix? yes
Free blocks count wrong for group #9 (32229, counted=2965).
Fix? yes
Free blocks count wrong for group #10 (32768, counted=5881).
Fix? yes
Free blocks count wrong for group #11 (32768, counted=3126).
Fix? yes
Free blocks count wrong for group #12 (32768, counted=3964).
Fix? yes
Free blocks count wrong for group #13 (32768, counted=4745).
Fix? yes
Free blocks count wrong for group #14 (32768, counted=4077).
Fix? yes
Free blocks count wrong for group #15 (16384, counted=0).
Fix? yes
Free blocks count wrong for group #16 (24624, counted=19419).
Fix? yes
Free blocks count wrong for group #17 (32768, counted=2).
Fix? yes
Free blocks count wrong for group #18 (32768, counted=0).
Fix? yes
Free blocks count wrong for group #19 (32768, counted=0).
Fix? yes
Free blocks count wrong for group #20 (32768, counted=3665).
Fix? yes
Free blocks count wrong for group #21 (32768, counted=1549).
Fix? yes
Free blocks count wrong for group #22 (32768, counted=2784).
Fix? yes
Free blocks count wrong for group #23 (32768, counted=3529).
Fix? yes
Free blocks count wrong for group #24 (32768, counted=5204).
Fix? yes
Free blocks count wrong for group #25 (32229, counted=4288).
Fix? yes
Free blocks count wrong for group #26 (32768, counted=32413).
Fix? yes
Free blocks count wrong (1063560, counted=378276).
Fix? yes
Free inodes count wrong for group #0 (8101, counted=7).
Fix? yes
Directories count wrong for group #0 (2, counted=1496).
Fix? yes

```

```

Free inodes count wrong for group #1 (8112, counted=0).
Fix? yes
Directories count wrong for group #1 (0, counted=287).
Fix? yes
Free inodes count wrong for group #2 (8112, counted=0).
Fix? yes
Directories count wrong for group #2 (0, counted=578).
Fix? yes
Free inodes count wrong for group #3 (8112, counted=0).
Fix? yes
Directories count wrong for group #3 (0, counted=492).
Fix? yes
Free inodes count wrong for group #4 (8112, counted=6376).
Fix? yes
Directories count wrong for group #4 (0, counted=287).
Fix? yes
Free inodes count wrong for group #16 (8112, counted=0).
Fix? yes
Directories count wrong for group #16 (0, counted=961).
Fix? yes
Free inodes count wrong for group #17 (8112, counted=0).
Fix? yes
Directories count wrong for group #17 (0, counted=538).
Fix? yes
Free inodes count wrong for group #18 (8112, counted=0).
Fix? yes
Free inodes count wrong for group #19 (8112, counted=0).
Fix? yes
Directories count wrong for group #19 (0, counted=437).
Fix? yes
Free inodes count wrong for group #20 (8112, counted=0).
Fix? yes
Directories count wrong for group #20 (0, counted=772).
Fix? yes
Free inodes count wrong for group #21 (8112, counted=0).
Fix? yes
Directories count wrong for group #21 (0, counted=554).
Fix? yes
Free inodes count wrong for group #22 (8112, counted=0).
Fix? yes
Directories count wrong for group #22 (0, counted=522).
Fix? yes
Free inodes count wrong for group #23 (8112, counted=3231).
Fix? yes
Directories count wrong for group #23 (0, counted=498).
Fix? yes
Free inodes count wrong for group #24 (8112, counted=7934).
Fix? yes
Directories count wrong for group #24 (0, counted=178).
Fix? yes
Free inodes count wrong (275797, counted=179788).
Fix? yes
Inode bitmap differences: Group 0 inode bitmap does not match checksum.
FIXED.

/dev/sda1: ***** FILE SYSTEM WAS MODIFIED *****
/dev/sda1: 96020/275808 files (0.1% non-contiguous), 723292/1101568 blocks

```

# umount /media/recuperacion #Desmontar (deixar de facer uso) a partición primaria /dev/sdb1 que estaba montada en /media/recuperacion

### Comprobar de novo o arranque do sistema operativo.

15. Apagar a máquina virtual.

**VBoxManage controlvm Debian32-Instalacion poweroff**

16. Modificar na configuración da máquina virtual as opcións de arranque e pór como primeira opción o Disco duro Virtual.

17. Arrancar de novo a máquina virtual creada en modo Inicio normal

**VBoxManage startvm Debian32-Instalacion**

18. Comprobar o arranque do sistema operativo.

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/)

# Comandos apt, dpkg

## Práctica 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 show debian-refcard #Amosar detalles do paquete debian-refcard

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

# apt reinstall debian-refcard #Reinstalar o paquete debian-refcard

# apt remove debian-refcard #Eliminar o paquete debian-refcard

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

# 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

# apt policy debian-refcard #Amosa o posible paquete candidato de debian-refcard a instalar e, dado o caso, que versión do paquete debian-refcard está instalado.

# apt autoremove #Eliminar automaticamente tódolos paquetes que foron instalados para satisfacer dependencias para outros paquetes e xa non son necesarios, ben porque as dependencias cambiaron ou os paquetes que os necesitaban xa foron borrados

# apt -y upgrade #Actualizar o sistema instalando/actualizando paquetes respostando **yes** a tódalas posibles preguntas durante a actualización

# apt -y full-upgrade #Actualizar o sistema eliminando/instalando/actualizando paquetes respostando **yes** a tódalas posibles preguntas durante a actualización

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

## Práctica apt-get

\$ apt-get #Amosar o uso do comando apt-get

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

# apt-get update #Actualizar lista de paquetes a instalar dos repositorios

# apt-get -y upgrade #Actualizar o sistema (todos os ficheiros actualmente instalados) respostando **yes** a tódalas posibles preguntas durante a actualización

# apt-get -y install debian-refcard #Instalar o paquete debian-refcard respostando **yes** a tódalas posibles preguntas durante a instalación

# apt-get -y remove debian-refcard #Eliminar o paquete debian-refcard respostando **yes** a tódalas posibles preguntas durante a instalación

# apt-get -y install debian-refcard #Instalar o paquete debian-refcard respostando **yes** a tódalas posibles preguntas durante a instalación

# apt-get -y purge debian-refcard #Eliminar e purgar o paquete debian-refcard respostando **yes** a tódalas posibles preguntas durante a instalación. É equivalente á opción remove pero ademais elimina calquera arquivo de configuración do paquete

# apt-get clean #Limpar os ficheiros descargados no repositorio local, é dicir, elimina todos os ficheiros existentes nas rutas /var/cache/apt/archives/ e /var/cache/apt/archives/partial/

# apt-get autoclean #Similar ao comando anterior, pero soamente elimina os arquivos dos paquetes que xa non se poden descargar e levan tempo sen usar

# apt-get autoremove #Eliminar automaticamente tódolos paquetes que foron instalados para satisfacer dependencias para outros paquetes e xa non son necesarios, ben porque as dependencias cambiaron ou os paquetes que os necesitaban xa foron borrados

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

\$ mkdir /tmp/paquetes && cd /tmp/paquetes #Crear o cartafol /tmp/paquetes e no caso que o comando teña éxito, é dicir, sexa executado sen erros, farase o segundo comando o cal accede ao directorio /tmp/paquetes

\$ 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.

## Práctica apt-cache

\$ apt-cache #Amosar o uso do comando apt-cache. Este comando non precisa ser executado con permisos de root

\$ apt-cache policy debian-refcard #Amosa o posible paquete candidato de debian-refcard a instalar e, dado o caso, que versión do paquete debian-refcard está instalado.

\$ apt-cache search refcard #Buscar nas descrições dos paquetes o patrón refcard

\$ apt-cache show debian-refcard #Amosar detalles do paquete debian-refcard

\$ apt-cache depends debian-refcard #Amosar información de dependencias do paquete debian-refcard

\$ apt-cache depends ntp #Amosar información de dependencias do paquete ntp

\$ apt-cache rdepends debian-refcard #Amosar información de dependencias inversas do paquete debian-refcard

\$ apt-cache rdepends ntp #Amosar información de dependencias inversas do paquete ntp

\$ apt-cache depends debian-refcard ntp #Amosar información de dependencias dos paquetes debian-refcard e ntp

\$ apt-cache rdepends debian-refcard ntp #Amosar información de dependencias inversas dos paquetes debian-refcard e ntp

\$ apt-get update #Actualizar lista de paquetes a instalar dos repositorios. **Amosa erro porque este comando precisa ser executado con permisos de root.**

\$ apt-get -y install debian-refcard #Instalar o paquete debian-refcard respostando **yes** a tódalas posibles preguntas durante a instalación. **Amosa erro porque este comando precisa ser executado con permisos de root.**

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

# apt-get -y install debian-refcard terminator #Instalar os paquetes debian-refcard e terminator respostando **yes** a tódalas posibles preguntas durante a instalación

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

## Práctica dpkg

\$ dpkg --help #Amosar a axuda do comando dpkg

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

\$ dpkg -L debian-refcard #Listar os ficheiros pertencentes ao paquete debian-refcard

\$ dpkg -S /usr/share/doc/debian-refcard/refcard-es-a4.pdf.gz #Atopar o/s paquete/s propietario/s do/s ficheiro/s buscado/s.

\$ dpkg -l debian-refcard terminator #Listar información sobre os paquetes debian-refcard e terminator

```
$ dpkg -L debian-refcard terminator #Listar os ficheiros pertencentes aos paquetes debian-refcard e terminator
```

```
$ dpkg -S /usr/share/doc/debian-refcard/refcard-es-a4.pdf.gz /usr/bin/terminator #Atopar o/s paquete/s propietario/s do/s ficheiro/s buscado/s.
```

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

```
# dpkg -r debian-refcard #Eliminar o paquete debian-refcard
```

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

```
# cd /tmp/paquetes #Cambiar ao directorio /tmp/paquetes
```

```
# 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
```

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

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

```
$
```

Ricardo Feijoo Costa



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