

Repaso Linux Práctica 3

PROCESOS Y COMPRIMIR

Envío a segundo plano y eliminación de procesos:

El comando `yes` envía una serie de 'y' a la salida estándar, `/dev/null` es un dispositivo 'agujero negro' todo lo que se le pase 'desaparecerá', así '`yes > /dev/null`' no mostrara nada en la salida estándar ya que todo se va a `/dev/null`. Para enviar un proceso a segundo plano lo llamamos desde la línea de comandos seguido de '&'.
1. Ejecuta el comando `yes > /dev/nul` y ejecuta `ctrl +c` para parar el proceso

```
root@ubuntu18sor-VirtualBox:/home/ubuntu18sor# yes > /dev/null
^C
root@ubuntu18sor-VirtualBox:/home/ubuntu18sor#
```

2. Vuelve a ejecutar `yes > /dev/null &`. ¿Qué es el número que va entre corchetes [nº] y el número que lo sigue?

```
root@ubuntu18sor-VirtualBox:/home/ubuntu18sor# yes > /dev/null &
[3] 2338
```

3. A continuación, ejecuta el comando `Jobs`. ¿En qué estado está el proceso?

```
root@ubuntu18sor-VirtualBox:/home/ubuntu18sor# jobs
[1]-  Stopped                  yes < /dev/sdb
[2]+  Stopped                  yes < /dev/null
[3]   Running                  yes > /dev/null &
```

El proceso está corriendo

4. Ahora ejecuta el comando `ps`. ¿Cuánto tiempo lleva el proceso ejecutándose?

```
root@ubuntu18sor-VirtualBox:/home/ubuntu18sor# ps
  PID TTY          TIME CMD
 2296 pts/0        00:00:00 sudo
 2297 pts/0        00:00:00 su
 2310 pts/0        00:00:00 bash
 2321 pts/0        00:00:00 yes
 2330 pts/0        00:00:00 yes
 2338 pts/0        00:02:46 yes
 2342 pts/0        00:00:00 ps
```

5. Vamos a matar el proceso (terminar) con el comando kill. Procede a matar el proceso puedes utilizar el número de tarea o si identificador de proceso. Elige una y terminar el proceso yes.

```
root@ubuntu18sor-VirtualBox:/home/ubuntu18sor# kill 2338
root@ubuntu18sor-VirtualBox:/home/ubuntu18sor# ps
  PID TTY          TIME CMD
 2296 pts/0    00:00:00 sudo
 2297 pts/0    00:00:00 su
 2310 pts/0    00:00:00 bash
 2321 pts/0    00:00:00 yes
 2330 pts/0    00:00:00 yes
 2346 pts/0    00:00:00 ps
[3] Terminated                  yes > /dev/null
```

6. Ejecuta jobs para ver que nos dice sobre el proceso.

```
root@ubuntu18sor-VirtualBox:/home/ubuntu18sor# jobs
[1]-  Stopped                  yes < /dev/sdb
[2]+  Stopped                  yes < /dev/null
```

Parada y relanzamiento de tareas:

1. Para poner una tarea en segundo plano después de haberla lanzada usamos ctrl +z. Ejecuta yes > /dev/null y a continuación pulsa CTRL + Z.

```
root@ubuntu18sor-VirtualBox:/home/ubuntu18sor# yes > /dev/null
^Z
[3]+  Stopped                  yes > /dev/null
```

2. Vamos a traer a primer plano el proceso yes utilizando el comando fg y volvemos a pulsar CTRL + Z

```
root@ubuntu18sor-VirtualBox:/home/ubuntu18sor# fg
yes > /dev/null
^Z
[3]+  Stopped                  yes > /dev/null
```

3. Ahora lo mandamos a segundo plano utilizando el comando bg y comprobar con jobs en qué estado se encuentra.

```
root@ubuntu18sor-VirtualBox:/home/ubuntu18sor# jobs
[1]-  Stopped                  yes < /dev/sdb
[2]+  Stopped                  yes < /dev/null
[3]   Running                  yes > /dev/null &
```

4. Lo volvemos a traer a primer plano y lo volvemos a parar con CTRL +Z.

```
root@ubuntu18sor-VirtualBox:/home/ubuntu18sor# fg
yes > /dev/null
^Z
[4]+  Stopped                  yes > /dev/null
```

5. Mata el proceso utilizando el número de tarea.

```
root@ubuntu18sor-VirtualBox:/home/ubuntu18sor# kill %2
[2]-  Stopped                  yes < /dev/null
```

6. Vamos a lanzar el proceso `yes > /dev/null &` dos veces, esto ejecutará el proceso en segundo plano dos veces. Y de nuevo lanzamos el proceso `yes > /dev/null`, lanzamos este proceso en primer plano y después de lanzarlo lo paramos con CTRL + Z sólo este último proceso.

```
root@ubuntu18sor-VirtualBox:/home/ubuntu18sor# yes > /dev/null &
[5] 2365
root@ubuntu18sor-VirtualBox:/home/ubuntu18sor# yes > /dev/null &
[6] 2366
root@ubuntu18sor-VirtualBox:/home/ubuntu18sor# yes > /dev/null
^Z
[7]+  Stopped                  yes > /dev/null
root@ubuntu18sor-VirtualBox:/home/ubuntu18sor#
```

7. Ejecutamos el comando `jobs` y vemos en qué estado están los procesos lanzados.

```
ubuntu18sor@ubuntu18sor-VirtualBox:~$ jobs
[1]  Running                  yes > /dev/null &
[2]-  Running                  yes > /dev/null &
[3]+  Stopped                  yes > /dev/null
```

8. Ejecutamos el comando `bg` que proceso lanza a segundo plano.

```
ubuntu18sor@ubuntu18sor-VirtualBox:~$ bg %3
[3]+ yes > /dev/null &
```

9. Ahora vamos a pulsar %nº esto trae a primer plano el proceso indicado en el número. Trae al primer plano el proceso con número de tarea 2. Pulsa CTRL + Z. Y ejecuta el comando jobs para estado de los procesos.

```
ubuntu18sor@ubuntu18sor-VirtualBox:~$ fg %2
yes > /dev/null
^Z
[2]+  Stopped                  yes > /dev/null
```

10. Mata el proceso con número tarea 2. Ejecuta jobs y comprueba los estados de los procesos.

```
ubuntu18sor@ubuntu18sor-VirtualBox:~$ kill %2
[2]+  Stopped                  yes > /dev/null
ubuntu18sor@ubuntu18sor-VirtualBox:~$ jobs
[1]   Running                  yes > /dev/null &
[2]+  Stopped                  yes > /dev/null
[3]-  Running                  yes > /dev/null &
```

11. A continuación, elimina todos los procesos que todavía están lanzados.

```
ubuntu18sor@ubuntu18sor-VirtualBox:~$ kill %1
ubuntu18sor@ubuntu18sor-VirtualBox:~$ kill %3
[1]   Terminated              yes > /dev/null
[2]+  Terminated              yes > /dev/null
ubuntu18sor@ubuntu18sor-VirtualBox:~$ jobs
[3]+  Terminated              yes > /dev/null
```


Prioridad de ejecución:

1. El comando nice indica la prioridad del proceso a la hora de su creación puede ir de -20 (prioridad más baja) a 19 (prioridad más alta). Valor por defecto es cero. Ejecuta nice para ver que prioridad tiene los proceso.

```
ubuntu18sor@ubuntu18sor-VirtualBox:~$ nice
0
```

2. Para cambiar prioridad de un proceso ejecutamos renice. Vamos a abrir el editor de texto gedit.

```
ubuntu18sor@ubuntu18sor-VirtualBox:~$ nice gedit
```

3. Paramos el editor pulsar CTRL + Z. Y ejecutamos ps -l. ¿Qué prioridad tiene gedit?

```
ubuntu18sor@ubuntu18sor-VirtualBox:~$ ps -l
F S  UID  PID  PPID  C PRI  NI ADDR SZ WCHAN  TTY          TIME CMD
0 S  1000  1642  1632  0  80   0 -  7487 wait  pts/0    00:00:00 bash
0 T  1000  1796  1642  1  90  10 - 169391 signal pts/0    00:00:00 gedit
0 R  1000  1806  1642  0  80   0 -  9004 -      pts/0    00:00:00 ps
```

Tiene una prioridad de 10.

4. Vamos a cambiar su prioridad con renice. Ejecutando renice 1 pid (número identificador del proceso). Vuelve a ejecutar ps -l

```
root@ubuntu18sor-VirtualBox:/home/ubuntu18sor# renice 1 1796
1796 (process ID) old priority 10, new priority 1
```

5. Ahora vamos a ejecutar nice -n 2 gedit. Ejecuta de nuevo ps -l y indica qué prioridad tiene el editor abierto ahora.

```
root@ubuntu18sor-VirtualBox:/home/ubuntu18sor# nice -n 2 gedit
^Z
[1]+  Stopped                  nice -n 2 gedit
root@ubuntu18sor-VirtualBox:/home/ubuntu18sor# ps -l
F S  UID  PID  PPID  C PRI  NI ADDR SZ WCHAN  TTY          TIME CMD
4 S   0  1813  1642  0  80   0 -  18239 poll_s pts/0    00:00:00 sudo
4 S   0  1814  1813  0  80   0 -  18060 wait  pts/0    00:00:00 su
4 S   0  1827  1814  0  80   0 -   7144 wait  pts/0    00:00:00 bash
4 T   0  1838  1827  0  82   2 -  99898 signal pts/0    00:00:00 gedit
0 R   0  1872  1827  0  80   0 -   9004 -      pts/0    00:00:00 ps
```

6. Otra forma de ver procesos es con el comando `pstree`. Ejecuta dicho comando y localiza los procesos `gedit` que tenemos lanzados.

```

gnome-terminal- bash- gedit-4*[{gedit}]
                  |
                  | sudo- su- bash- gedit-2*[{ged+
                  |                  |
                  |                  | pstree
                  |
                  | 2*[{gnome-terminal- 11

```

7. Ahora vamos a ejecutar el comando `top`, muestra también los procesos en tiempo real y el tiempo de cpu y memoria que están gastando.

```

top - 22:40:18 up 26 min,  1 user,  load average: 0,23, 0,21, 0,39
Tasks: 188 total,  1 running, 150 sleeping,  2 stopped,  0 zombie
%Cpu(s):  9,6 us,  4,0 sy,  0,0 ni, 86,4 id,  0,0 wa,  0,0 hi,  0,0 si,  0,0 st
KiB Mem : 4030896 total, 2296216 free,  814220 used,  920460 buff/cache
KiB Swap: 483800 total, 483800 free,    0 used. 2944148 avail Mem

```

8. Lanza el proceso `yes` en segundo plano como ya se ha hecho en la práctica y ejecuta el comando `jobs`. Mira los procesos que están en ejecución y parados.

```

root@ubuntu18sor-VirtualBox:/home/ubuntu18sor# jobs
[1]-  Stopped                  nice -n 2 gedit
[2]+  Stopped                  top
[3]   Running                  yes > /dev/null &

```

9. Ejecuta el comando `top -U alumno` solo saldrán los procesos de usuarios. Sin cerrar `top` escribe `u` y cambia al usuario `root`. Vuelve a cambiar al usuario `alumno`. Y ahora pulsa `k` e indica el pid del proceso `yes` para que sea matado. Para salir de `top` pulsa `q`.

```

ubuntu18sor@ubuntu18sor-VirtualBox:~$ top -U ubuntu18sor
top - 22:45:25 up 31 min,  1 user,  load average: 1,22, 0,75, 0,56
Tasks: 186 total,  3 running, 146 sleeping,  1 stopped,  0 zombie
%Cpu(s): 34,2 us, 65,8 sy,  0,0 ni,  0,0 id,  0,0 wa,  0,0 hi,  0,0 si,  0,0 st
KiB Mem : 4030896 total, 2303824 free,  806572 used,  920500 buff/cache
KiB Swap: 483800 total, 483800 free,    0 used. 2951820 avail Mem

```

```

Which user (blank for all) root
  PID USER      PR  NI   VIRT    RES    SHR S  %CPU  %MEM    TIME+  COMMAND
 1590 root        20   0  577296  22544  19488 S   0,0   0,6   0:00.14 fwupd
 1815 root        20   0   76960   8164   6780 S   0,0   0,2   0:00.10 systemd
 1816 root        20   0  259060   2528    40 S   0,0   0,1   0:00.00 (sd-pam)
 1840 root        20   0   49924   3824   3464 S   0,0   0,1   0:00.02 dbus-daemon
 1842 root        20   0  291960   7156   6364 S   0,0   0,2   0:00.01 gvfsd
 1847 root        20   0  366480   8156   7216 S   0,0   0,2   0:00.01 gvfsd-fuse
 1858 root        20   0  429856   6204   5640 S   0,0   0,2   0:00.00 xdg-documen+
 1861 root        20   0  271560   5264   4828 S   0,0   0,1   0:00.00 xdg-permiss+
 1876 root        20   0         0         0     0 I   0,0   0,0   0:00.04 kworker/u2:+
 1906 root        20   0         0         0     0 I   0,0   0,0   0:00.00 kworker/u2:+

```

NOMBRE ALUMNO:

```

kib swap: 485888 total, 0 485888 free, 0 used. 2951888 avail mem
PID to signal/kill [default pid = 1529] 1529

```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
-----	------	----	----	------	-----	-----	---	------	------	-------	---------

10. Ejecuta jobs y mata todos los procesos que están todavía abiertos

```

ubuntu18sor@ubuntu18sor-VirtualBox:~$ ps -l
F S      UID      PID  PPID  C PRI  NI ADDR SZ WCHAN  TTY          TIME CMD
0 S      1000     1642   1632  0  80   0 -  7487 wait    pts/0        00:00:00 bash
0 T      1000     1796   1642  0  81   1 - 169391 signal pts/0        00:00:00 gedit
0 T      1000     1912   1642  0  80   0 - 12795 signal pts/0        00:00:00 top
0 R      1000     1915   1642  0  80   0 -  9004 -      pts/0        00:00:00 ps
ubuntu18sor@ubuntu18sor-VirtualBox:~$ kill 1642
ubuntu18sor@ubuntu18sor-VirtualBox:~$ kill 1796
ubuntu18sor@ubuntu18sor-VirtualBox:~$ kill 1912
ubuntu18sor@ubuntu18sor-VirtualBox:~$ kill 1915

```


Manejo de procesos:

1. Determinar el UID efectivo de la sesión actual. Determinar el PID (Process ID) del proceso init.

```
ubuntu18sor@ubuntu18sor-VirtualBox:~$ ps axu
```

USER	PID	%CPU	%MEM	VSZ	RSS	TTY	STAT	START	TIME	COMMAND
root	1	0.0	0.2	225284	9020	?	Ss	22:14	0:02	/sbin/init splash

2. Ejecutar en background (segundo plano) el comando

```
/dev/null
```

```
ubuntu18sor@ubuntu18sor-VirtualBox:~$ ping 8.8.8.8 > /dev/null &
[3] 2039
```

3. Ejecutar el comando, una vez iniciado enviarlo a segundo plano.

```
[4]+  Stopped                  top
ubuntu18sor@ubuntu18sor-VirtualBox:~$
```

4. Mostrar todos los procesos que se están ejecutando con el mismo UID efectivo que la sesión actual

```
ubuntu18sor@ubuntu18sor-VirtualBox:~$ ps -f
```

UID	PID	PPID	C	STIME	TTY	TIME	CMD
ubuntu1+	1642	1632	0	22:16	pts/0	00:00:00	bash
ubuntu1+	1912	1642	0	22:45	pts/0	00:00:00	top -U ubuntu18sor
ubuntu1+	2039	1642	0	23:08	pts/0	00:00:00	ping 8.8.8.8
ubuntu1+	2042	1642	0	23:09	pts/0	00:00:00	top
ubuntu1+	2043	1642	0	23:10	pts/0	00:00:00	ps -f

5. Mostrar un árbol sólo de los procesos que se están ejecutando con el mismo UID efectivo que la sesión actual

```
ubuntu18sor@ubuntu18sor-VirtualBox:~$ ps -u ubuntu18sor
```

PID	TTY	TIME	CMD
992	?	00:00:00	systemd

6. Mostrar todos los procesos ejecutando como usuario "root" (ID efectivo y real) en formato de usuario

```
ubuntu18sor@ubuntu18sor-VirtualBox:~$ sudo su
[sudo] password for ubuntu18sor:
root@ubuntu18sor-VirtualBox:/home/ubuntu18sor# ps -f
```

UID	PID	PPID	C	STIME	TTY	TIME	CMD
root	1905	992	98	22:40	pts/0	00:31:21	yes
root	2046	1642	0	23:12	pts/0	00:00:00	sudo su
root	2048	2047	0	23:12	pts/0	00:00:00	bash
root	2060	2048	0	23:12	pts/0	00:00:00	ps -f

7. Determinar el PID (Process ID) del proceso ejecutando el comando

```
1912 pts/0    00:00:00 top
2039 pts/0    00:00:00 ping
2042 pts/0    00:00:00 top
```

8. Bajar al mínimo la prioridad de ejecución del proceso ejecutando el comando `ping 8.8.8.8 > /dev/null`

```
ubuntu18sor@ubuntu18sor-VirtualBox:~$ sudo renice -n 20 2039
2039 (process ID) old priority 0, new priority 19
```

La prioridad más baja disponible es 19

9. Matar el proceso ejecutando el comando `ping 8.8.8.8 > /dev/null`

```
ubuntu18sor@ubuntu18sor-VirtualBox:~$ kill 2039
```

10. Determinar qué proceso está consumiendo más CPU

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
1905	root	20	0	14572	772	708	R	98,7	0,0	37:46.09	yes
1196	ubuntu1+	20	0	3010080	280684	104924	S	0,3	7,0	1:12.82	/usr/bin/gn+
1632	ubuntu1+	20	0	801540	37496	28176	S	0,3	0,9	0:11.68	/usr/lib/gn+
2084	ubuntu1+	20	0	51208	4072	3368	R	0,3	0,1	0:03.98	top
1	root	20	0	225284	9020	6676	S	0,0	0,2	0:02.60	/sbin/init +
2	root	20	0	0	0	0	S	0,0	0,0	0:00.00	[kthreadd]

11. Determinar qué proceso está consumiendo más memoria RAM

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
1905	root	20	0	14572	772	708	R	98,7	0,0	37:46.09	yes
1196	ubuntu1+	20	0	3010080	280684	104924	S	0,3	7,0	1:12.82	/usr/bin/gn+
1632	ubuntu1+	20	0	801540	37496	28176	S	0,3	0,9	0:11.68	/usr/lib/gn+
2084	ubuntu1+	20	0	51208	4072	3368	R	0,3	0,1	0:03.98	top
1	root	20	0	225284	9020	6676	S	0,0	0,2	0:02.60	/sbin/init +
2	root	20	0	0	0	0	S	0,0	0,0	0:00.00	[kthreadd]

12. Determinar cuanta memoria RAM disponible (libre) posee el sistema

```
top - 23:21:51 up 1:07, 1 user, load average: 1,04, 1,10, 1,04
Tasks: 182 total, 2 running, 143 sleeping, 2 stopped, 0 zombie
%Cpu(s): 35,6 us, 64,4 sy, 0,0 ni, 0,0 id, 0,0 wa, 0,0 hi, 0,0 si, 0,0 st
KiB Mem : 4030896 total, 2455716 free, 658224 used, 916956 buff/cache
KiB Swap: 483800 total, 483800 free, 0 used. 3104812 avail Mem
```

13. Traer a primer plano el proceso ejecutando el comando `top`

```
ubuntu18sor@ubuntu18sor-VirtualBox:~$ jobs
[2] Stopped top -U ubuntu18sor
[4]- Stopped top
[5]+ Stopped top
ubuntu18sor@ubuntu18sor-VirtualBox:~$ fg %2
top -U ubuntu18sor
```

14. Comando tar , gzip, filtros, direccionar y df:

1. Crear un directorio llamado "docA" que contenga los archivos "doc1.txt", "doc3.txt" y "doc77.txt". Crear un directorio llamado "docB" que contenga los archivos "doc1.txt", "doc4.txt" y "doc89.txt".

```
ubuntu18sor@ubuntu18sor-VirtualBox:~/practicass$ ls
ubuntu18sor@ubuntu18sor-VirtualBox:~/practicass$ mkdir docA
ubuntu18sor@ubuntu18sor-VirtualBox:~/practicass$ mkdir docB
ubuntu18sor@ubuntu18sor-VirtualBox:~/practicass$ cd docA
ubuntu18sor@ubuntu18sor-VirtualBox:~/practicass/docA$ > doc1.txt
ubuntu18sor@ubuntu18sor-VirtualBox:~/practicass/docA$ > doc3.txt
ubuntu18sor@ubuntu18sor-VirtualBox:~/practicass/docA$ > doc77.txt
ubuntu18sor@ubuntu18sor-VirtualBox:~/practicass/docA$ cd ..
ubuntu18sor@ubuntu18sor-VirtualBox:~/practicass$ cd docB
ubuntu18sor@ubuntu18sor-VirtualBox:~/practicass/docB$ > doc1.txt
ubuntu18sor@ubuntu18sor-VirtualBox:~/practicass/docB$ > doc4.txt
ubuntu18sor@ubuntu18sor-VirtualBox:~/practicass/docB$ > doc89.txt
ubuntu18sor@ubuntu18sor-VirtualBox:~/practicass/docB$ cd ..
```

2. Listar, utilizando una única línea de comandos, los nombres de archivo que se encuentran en el directorio "docA" pero no en el directorio "docB"

```
ubuntu18sor@ubuntu18sor-VirtualBox:~$ ls /home/ubuntu18sor/practicass/docA
doc1.txt doc3.txt doc77.txt
```

3. Listar, utilizando una única línea de comandos, los nombres de archivo que se encuentran tanto en el directorio "docA" como en el directorio "docB"

```
ubuntu18sor@ubuntu18sor-VirtualBox:~$ ls /home/ubuntu18sor/practicass/docA ; ls /
home/ubuntu18sor/practicass/docB
doc1.txt doc3.txt doc77.txt
doc1.txt doc4.txt doc89.txt
```

4. Comprimir los directorios "docA" y "docB" utilizando el comando "gzip" en un único archivo llamado "docab.tar.gz".

Utilizaré el comando **tar** ya que no es posible comprimir directorios con **gzip**

```
ubuntu18sor@ubuntu18sor-VirtualBox:~$ tar -cvf docab.tar /home/ubuntu18sor/pract
icas ; gzip docab.tar
tar: Removing leading '/' from member names
/home/ubuntu18sor/practicass/
/home/ubuntu18sor/practicass/docA/
/home/ubuntu18sor/practicass/docA/doc77.txt
/home/ubuntu18sor/practicass/docA/doc1.txt
/home/ubuntu18sor/practicass/docA/doc3.txt
/home/ubuntu18sor/practicass/docB/
/home/ubuntu18sor/practicass/docB/doc1.txt
/home/ubuntu18sor/practicass/docB/doc89.txt
/home/ubuntu18sor/practicass/docB/doc4.txt
```


5. Volcar el contenido del archivo `/etc/fstab` exceptuando las líneas que contengan el caracter '#'.

```
ubuntu18sor@ubuntu18sor-VirtualBox:~$ grep -v "#" /etc/fstab > Archivo1.txt
ubuntu18sor@ubuntu18sor-VirtualBox:~$ ls
Archivo1.txt  examples.desktop  nano.save  Pictures  Public  Templates
Desktop      gastos            paises     piezas.Lopez  script1  Videos
docab.tar.gz misfotos          paises.txt piezas.Perez  telef    Windows
Documents    multimedia        Panama     piezas.Ruiz   telef1234
Downloads    Music             perm1      practicas    telef134
ubuntu18sor@ubuntu18sor-VirtualBox:~$ cat Archivo1.txt
UUID=7ff50481-d5ee-499b-84a0-a2014720ccec /          ext4    errors=remount
-ro 0      1
/swapfile                                none    swap    sw
0          0
/home/COPIAS ext4 auto defaults 0 0
```

6. Volcar el contenido del archivo `/etc/fstab` exceptuando las 3 primeras líneas.

```
ubuntu18sor@ubuntu18sor-VirtualBox:~$ more +4 /etc/fstab > Archivo2.txt
ubuntu18sor@ubuntu18sor-VirtualBox:~$ cat Archivo2.txt
# device; this may be used with UUID= as a more robust way to name devices
# that works even if disks are added and removed. See fstab(5).
#
# <file system> <mount point> <type> <options>          <dump> <pass>
# / was on /dev/sda1 during installation
UUID=7ff50481-d5ee-499b-84a0-a2014720ccec /          ext4    errors=remount
-ro 0      1
/swapfile                                none    swap    sw
0          0
/home/COPIAS ext4 auto defaults 0 0
```

7. Buscar recursivamente dentro del directorio `/etc` todos los archivos cuyo nombre comienza con "net".

```
ubuntu18sor@ubuntu18sor-VirtualBox:~$ find /etc/net*
/etc/netplan
/etc/netplan/01-network-manager-all.yaml
/etc/network
```

8. Determinar cuánto espacio en disco ocupa el directorio `/home` expresados en KB, MB o GB dependerá del tamaño que tengamos.

```
ubuntu18sor@ubuntu18sor-VirtualBox:~$ df -H /home
Filesystem      Size  Used Avail Use% Mounted on
/dev/sda1       11G   6,4G  3,7G  64% /
```