UADE

Trabajo Práctico I

Tutorial de introducción a Linux

Sistemas Operativos

Group N° 16

Alumno	Legajo	Email
Cantero, Lucas Alejandro	1116559	studentA@uade.edu.ar
Ledesma, Eric	1119065	studentA@uade.edu.ar
Nicolino, Sebastian	1130322	studentB@uade.edu.ar
Rios Burgoa Gabriel Yamil	1129370	griosburgoa@uade.edu.ar

Facultad de Ingeniería y Ciencias Exactas
Universidad Argentina de la Empresa
Lima 775 (Campus Montserrat) - C1073AA0
Ciudad Autónoma de Buenos Aires - Rep. Argentina
Tel: 0800-122-8233 / (+54 11) 5892-1783
http://www.uade.edu.ar

Índice

1. Ayuda

1.1. Comando man

1.2. Comando whatis

1.3. Comando whereis

1.4. Comando help

1.5. Comando apropos

1.6. Comando info

2. Teclado / Terminales

3. Sistema

Investigue los comandos:

- **poweroff**

  Comando utilizado para apagar la máquina, este comando no requiere permisos de súper usuario.

- **reboot**

  Comando utilizado para reiniciar la máquina, este comando no requiere permisos de súper usuario.

Tanto poweroff como reboot son alias, enlaces simbolicos, del comando halt, comando que permite detener las funciones del CPU de la máquina.

4. Usuarios

« ¿Qué es la cuenta de superusuario (root) y para qué se utiliza? »

Las cuestas de superusuario o root son cuentas que poseen permisos de administrador, es decir, tienen priviligios y permisos para realizar acciones sobre el sistema. Contar con estos permisos permitirá ejecutar instrucciones sumamente utiles las cuales pueden tener un impacto positivo o negativo sobre nuestro sistema.

« Investigue qué sucede con la cuenta de root en Ubuntu (el sistema operativo de uso en la cátedra). Investigue cómo realizar comandos a nombre del usuario root. »

Como se mencionó anteriormente este comando le permite hacer cambios en el sistema, como por ejemplo, instalar una nueva aplicación con «sudo apt-get install your_favorite_app ». Para ejecutar instrucciones a nombre de usuario root se puede hacer de dos formas:

- **su:** comportamiento default de este comando permite **cambiar el usuario actual por un usuario root dentro de la misma sesión dentro de la terminal** y así ejecutar cuantas instrucciones necesitemos.

```
~uade_workplace/ssoo/tp1
grios@personal-pc:~$ su
Password:

root@personal-pc:~# exit

root@personal-pc:~$
```

- **sudo:** El comando sudo tiene un comportamiento similar al anterior, pero nos permite ejecutar instrucciones sin necesidad de iniciar o cambiar de usuario.

```
~uade_workplace/ssoo/tp1
grios@personal-pc:~$ sudo apt-get install {your-favorite-package}
[sudo] password for grios:
```

« Investigue los comandos: »

- **su:** Como se mencionó anteriormente este comando permite loggearse, dentro de una sesión de terminal, con un usuario con máximos privilegios.

```
~uade_workplace/ssoo/tp1
grios@personal-pc:~$ su
Password:
root@personal-pc:~#
```

- **login:** Es un comando que permite iniciar sesión, dentro del terminal, como otro usuario (puede ser root o no). Para ejecutar este comando es necesarios ejecutarlo con privilegios de root.

  La forma analoga a ejecutar el comando su sería:

```
~uade_workplace/ssoo/tp1
grios@personal-pc:~$ sudo login
[sudo] password for grios:
grios-pc login: root
Password:
Welcome to Ubuntu 20.04 LTS (GNU/Linux 5.4.0-42-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage


237 updates can be installed immediately.
0 of these updates are security updates.
To see these additional updates run: apt list --upgradable

Failed to connect to https://changelogs.ubuntu.com/meta-release-lts. Check your Internet connection or proxy

Your Hardware Enablement Stack (HWE) is supported until April 2025.
Last login: dom ago 23 00:56:08 -03 2020 on pts/0

root@personal-pc:~#
```

« Investigue qué hace el comando: »

- **adduser:** Nos permite crear cuentas de usuario. Durante el alta de usuario tenemos la posibilidad de configurar un directorio home y asignar un interprete de comandos, por lo general es /bin/bash. Una vez creado el usuario nos podemos loggear y configurar una contraseña
- **addgroup:** Nos permite crear un grupo indicando el nombre por parámetro.

Ambos comandos necesitan privilegios de superusuario para ser ejecutados.
« Cree un nuevo usuario, cree un nuevo grupo, y agregue el usuario a ese grupo. »

```
~uade_workplace/ssoo/tp1
grios@personal-pc:~$ su
Password:

root@personal-pc:~# useradd test_user
root@personal-pc:~# groupadd test_group
root@personal-pc:~# usermod -a -G test_group test_user

root@personal-pc:~#  cat /etc/passwd | grep 'test_user'
test_user:x:1001:1001::/home/test_user:/bin/sh

root@personal-pc:~# cat /etc/group | grep 'test_user'
test_user:x:1001:
test_group:x:1002:test_user

root@personal-pc:~# exit
```

La instrucción **usermod** permite modificar la información de las cuentas de usuario preexistentes, el flag **-a -G** permite agregar al usuario a un grupo (append group)
« Investigue qué hace el comando: »

- **deluser:** Este comando permite quitar usuarios de un grupo.

```
~uade_workplace/ssoo/tp1
grios@personal-pc:~$ sudo deluser {your_favorite_user} {your_favorite_group}
```

- **delgroup:** Este comando permite eliminar un grupo.

```
grios@personal-pc:~$ sudo delgroup {your_favorite_group}
```

« Borre el usuario creado anteriormente (incluyendo el borrado de su directorio en home y todos sus archivos). »

```
~uade_workplace/ssoo/tp1
grios@personal-pc:~$ sudo deluser test_user --remove-home
[sudo] password for grios:
Looking for files to backup/remove ...
Removing user ‘test_user’ ...
Warning: group ‘test_user’ has no more members.
Done.
```

« Investigue cómo hacer para saber todos los grupos a los que pertenece un usuario. »
Para lograr esto encontramos 3 formas:

- **Viendo files del sistema:** En el archivo /etc/group se puede encontrar información relevante de todos los grupos creados en el sistema, y si usamos el comando grep podemos obtener información del grupo que nos interese, como por ejemplo los usuarios que pertenecen a él.

```
~uade_workplace/ssoo/tp1
grios@personal-pc:~$ cat /etc/group | grep 'test_user'
test_user:x:1001:
test_group:x:1002:test_user
```

- **Comando getent:** El comando getent nos listará todos los grupos existentes en nuestra máquina, si usamos la misma tactica que antes podemo hacer un grep y ver a qué grupo pertenece un usuario puntual.

```
~uade_workplace/ssoo/tp1
grios@personal-pc:~$ getent group | grep 'test_user'
test_group:x:1002:test_user
test_user:x:1001:
```

- **Comando groups:** El comando groups quizá es el más adecuado para averiguar todos los grupos a los que pertenece un usuario, la forma de ejecutarlo es:

```
~uade_workplace/ssoo/tp1
grios@personal-pc:~$ groups test_user
test_user : test_user test_group
```

5. Archivos

« ¿Qué hacen los siguientes comandos? »

- **cp:** Permite copiar un archivo o varios desde un directorio origen a un destino.

```
~uade_workplace/ssoo/tp1
grios@personal-pc:~$ cp {sourche_path} {destination_path}
```

- **mv:** Permite mover un archivo (o directorio) desde un directorio origen a uno de destino. Además de esta funcionalidad permite renombrar un archivo o directorio. Por ejemplo:
  - Move directorio

```
~uade_workplace/ssoo/tp1
grios@personal-pc:~$ mv folder_test /your/favorite/destination_path/
```

  - Move archivo

```
~uade_workplace/ssoo/tp1
grios@personal-pc:~$ mv test_file.txt /your/favorite/destination_path/
```

  - Renombrar directorio

```
~uade_workplace/ssoo/tp1
grios@personal-pc:~$ mv /your/favorite/directory/ /your/favorite/directory/ your_favorite_new_directory_
```

  - Renombrar archivos

```
~uade_workplace/ssoo/tp1
grios@personal-pc:~$ mv test_file.txt test_file_with_new_name.txt
```

- **rm:** Permite borrar archivos y si se usa recursivamente (flag -r) permite borrar directorios
  - Borrar directorio

```
~uade_workplace/ssoo/tp1
grios@personal-pc:~$ rm folder_test
```

  - Borrar archivo

```
~uade_workplace/ssoo/tp1
grios@personal-pc:~$ mv -r /your/favorite/directory/
```

- **scp:** Este comando nos permite realizar transferencias de archivos o directorios desde nuestra máquina local a servidores remoto, y también permite la transferencia entre servidores remotos.
  - Transferencia local a remoto

```
~uade_workplace/ssoo/tp1
grios@personal-pc:~$ scp test_file.txt test_user_a@domain.com:/home/test_user_a
```

  - Transferencia remoto a local

```
~uade_workplace/ssoo/tp1
grios@personal-pc:~$ scp test_user_a@domain.com:/home/test_user_a/test_file.txt /home/grios
```

  - Transferencia remoto a remoto

```
~uade_workplace/ssoo/tp1
grios@personal-pc:~$ scp test_user_a@domain.com:/home/test_user_a/test_file.txt test_user_b@domain.com:/
```

- **telnet:** Este comando permite acceder de manera remota a un servidor mediante el protocolo telnet y ejecutar comandos de manera remota, hoy por hoy este comando es reemplazado en muchos casos por ssh, dado que telnet no es considerado un método seguro de transferencia de datos.

```
~uade_workplace/ssoo/tp1
grios@personal-pc:~$ telnet {your_favorite_ip}:{your_favorite_port}
```

- **ssh:** Este comando tiene las mismas prestaciones que el comando telnet, pero con la difernecia que se usa un protocolo ssh estableciendo un canal seguro ya que las información se transporta encriptada

```
~uade_workplace/ssoo/tp1
grios@personal-pc:~$ ssh test_user@yourfavoriteserver.domain.com {your_favorite_command}
```

- **touch:** Este comando generalmente se usa para crear archivos vacíos o cambiar las tiempos de actualización de un archivo preexistente (sólo el tiempo de acceso y tiempo de modificación).

```
~uade_workplace/ssoo/tp1
grios@personal-pc:~$ touch test_file.txt
```

« A la hora de referirse a archivos, se puede usar tanto su dirección relativa (al directorio en el que se encuentra situado) o absoluta. Sitúese como root dentro del directorio /root. Luego copie el archivo .bashrc a la ruta absoluta /var/.bashrc. Ahora, mueva ese archivo desde esa dirección hasta /home/.bashrc sin desplazarse del directorio inicial (/root). »

```
~uade_workplace/ssoo/tp1
grios@personal-pc:~$ su
Password:
root@personal-pc:/home/grios# cd /root/
root@personal-pc:~# cp .bashrc /var/.bashrc
root@personal-pc:~# cp /var/.bashrc /home/.bashrc
root@personal-pc:~# exit

~uade_workplace/ssoo/tp1
grios@personal-pc~$ l /var | grep 'bashrc'
-rw-r--r--  1 root root     3,1K ago 23 12:55 .bashrc

~uade_workplace/ssoo/tp1
grios@personal-pc~$ l /home | grep 'bashrc'
-rw-r--r--  1 root  root  3,1K ago 23 12:56 .bashrc
```

6. Permisos

« Cree un archivo tipeando "ls >archivo". »

```
~uade_workplace/ssoo/tp1
grios@personal-pc:~$ ls
enunciado.pdf  Informe

~uade_workplace/ssoo/tp1
grios@personal-pc:~$ ls > archivo

~uade_workplace/ssoo/tp1
grios@personal-pc:~$ ls
archivo enunciado.pdf  Informe
```

« Tipee ls –l en dicho directorio: los primeros 10 caracteres corresponden a los permisos. Investigue como se estructuran los permisos de un archivo (puede tipear info y luego ir a la sección de "permisos de archivo" o "file permissions"). »

```
~uade_workplace/ssoo/tp1
grios@personal-pc:~$ ls -l
total 200
-rw-rw-r-- 1 grios grios     30 ago 23 18:59 archivo
-rw-rw-r-- 1 grios grios 194796 ago 23 02:48 enunciado.pdf
drwxrwxr-x 3 grios grios   4096 ago 23 19:01 Informe
```

Los permiso de un archivo se estructuran con 10 caracteres (10 bits), los cuales se ven representados de la siguiente forma:

9	8	7	6	5	4	3	2	1	0
-	r	w	x	r	w	x	r	w	x
archivo	permiso del usuario			permisos de grupo			permisos para otros		

Notese que el bit más significativo tiene el caracter '-', el cual indica que corresponde a un archivo.

Donde la primer fila corresponde a permisos básicos que son **lectura (r), escritura (w) y ejecución (x)**. Una vez mencinados estos 3 tipos de permisos podemos ver que estos se agrupan de a tercios, donde cada uno corresponde a **permisos de usuarios, grupos y otros**.

« Investigue que hacen esos comandos. »

- chmod El comando 'change mode' permite cambiar los permisos de acceso sobre archivos y directorios.
- chown El comando 'change owner' permite el propiertario de archivos o directorios.

« Haga que el archivo "archivo" creado anteriormente pueda ser modificado por cualquier usuario. »

```
~uade_workplace/ssoo/tp1
grios@personal-pc:~$ chmod o+rw archivo

~uade_workplace/ssoo/tp1
grios@personal-pc:~$ ls -l
total 200
-rw-rw-rw- 1 grios grios     30 ago 23 18:59 archivo
-rw-rw-r-- 1 grios grios 194796 ago 23 02:48 enunciado.pdf
drwxrwxr-x 3 grios grios   4096 ago 23 19:58 Informe
```

« Compruebe que logró el punto anterior logueandose en otra Terminal con otro usuario y modificando dicho archivo (tipeando nuevamente "ls >archivo"). »

```
~uade_workplace/ssoo/tp1
grios@personal-pc:~$ sudo useradd -m -d $HOME test_user

~uade_workplace/ssoo/tp1
grios@personal-pc:~$ sudo login test_user
Password:
Welcome to Ubuntu 20.04 LTS (GNU/Linux 5.4.0-42-generic x86_64)...

$ cd uade_workplace/ssoo/tp1
$ ls
archivo  enunciado.pdf  Informe
$ ls > archivo
$ ls -l
total 200
-rw-rw-rw- 1 grios grios     30 ago 23 20:44 archivo
-rw-rw-r-- 1 grios grios 194796 ago 23 02:48 enunciado.pdf
drwxrwxr-x 3 grios grios   4096 ago 23 19:58 Informe
```

« Loguéese con el usuario original y quite los todos los permisos del archivo (lectura, escritura y ejecución) a todos los usuarios distintos del dueño y de los que pertenecen al mismo grupo. Luego, haga que el nuevo dueño del archivo sea el otro usuario. »

```
~uade_workplace/ssoo/tp1
grios@personal-pc:~$ chmod o-rwx,g-rwx archivo

~uade_workplace/ssoo/tp1
grios@personal-pc:~$ ls -l
total 200
-rw------- 1 grios grios     30 ago 23 20:53 archivo
-rw-rw-r-- 1 grios grios 194796 ago 23 02:48 enunciado.pdf
drwxrwxr-x 3 grios grios   4096 ago 23 20:58 Informe


~uade_workplace/ssoo/tp1
grios@personal-pc:~$ chown test_user archivo
chown: changing ownership of 'archivo': Operation not permitted

~uade_workplace/ssoo/tp1
grios@personal-pc:~$ sudo chown test_user archivo

~uade_workplace/ssoo/tp1
grios@personal-pc:~$ ls -l
total 200
-rw------- 1 test_user grios     30 ago 23 20:53 archivo
-rw-rw-r-- 1 grios     grios 194796 ago 23 02:48 enunciado.pdf
drwxrwxr-x 3 grios     grios   4096 ago 23 20:58 Informe
```

« ¿Cómo haría para volver a poseer dicho archivo sin loguearse con el nuevo dueño del archivo? »

La forma de recuperar la propiedad de un archivo es ejecutando ejecutan el comando chown con permisos de superusuario, por ejemplo:

```
~uade_workplace/ssoo/tp1
grios@personal-pc:~$ ls -l
total 200
-rw------- 1 test_user grios     30 ago 23 20:53 archivo
-rw-rw-r-- 1 grios     grios 194796 ago 23 02:48 enunciado.pdf
drwxrwxr-x 3 grios     grios   4096 ago 23 20:58 Informe

~uade_workplace/ssoo/tp1
grios@personal-pc:~$ sudo chown grios archivo

~uade_workplace/ssoo/tp1
grios@personal-pc:~$ ls -l
total 204
-rw------- 1 grios grios     30 ago 23 20:53 archivo
-rw-rw-r-- 1 grios grios 194796 ago 23 02:48 enunciado.pdf
drwxrwxr-x 3 grios grios   4096 ago 23 20:58 Informe
```

« Investigue qué es el "SUID bit" (busque en "man chmod"). »

El SUID bit (setup ID bit) se indica que todo aquel que ejecute el archivo tendrá, durante la ejecución, los mismo privilegios que el usuario owner del archivo.

« Investigue como aplica la estructura de los permisos a los directorios. »

Al igual que los archivos, los permiso de un directorio se estructuran con 10 caracteres (10 bits), los cuales se ven representados de la siguiente forma:

9	8	7	6	5	4	3	2	1	0
d	r	w	x	r	w	x	r	w	x
directorio	permiso del usuario			permisos de grupo			permisos para otros		

Notese que el bit más significativo tiene el caracter 'd', el cual indica que corresponde a un directorio.

« Loguéese como root en otra Terminal y cree un directorio tipeando "mkdir /undir". »

```
~uade_workplace/ssoo/tp1
grios@personal-pc:~$ su
Password:

~uade_workplace/ssoo/tp1
grios@personal-pc:~$ ls -l
total 204
-rw------- 1 grios grios     30 ago 23 20:53 archivo
-rw-rw-r-- 1 grios grios 194796 ago 23 02:48 enunciado.pdf
drwxrwxr-x 3 grios grios   4096 ago 23 20:58 Informe

root@grios-pc:/home/grios/uade_workplace/ssoo/tp1# mkdir undir

root@grios-pc:/home/grios/uade_workplace/ssoo/tp1# ls -l
total 200
-rw-rw-r-- 1 grios grios 194796 ago 23 02:48 enunciado.pdf
drwxrwxr-x 3 grios grios   4096 ago 23 22:16 Informe
drwxr-xr-x 2 root  root    4096 ago 23 22:26 undir
```

« Haga que cualquier usuario tenga todo tipo de permisos sobre ese directorio. »

```
root@grios-pc:/home/grios/uade_workplace/ssoo/tp1# chmod o+rwx undir/

root@grios-pc:/home/grios/uade_workplace/ssoo/tp1# ls -l
total 200
-rw-rw-r-- 1 grios grios 194796 ago 23 02:48 enunciado.pdf
drwxrwxr-x 3 grios grios   4096 ago 23 22:16 Informe
drwxr-xrwx 2 root  root    4096 ago 23 22:26 undir
```

« Deshaga lo que acaba de hacer, y cree el subdirectorio “subdir” dentro de “/undir” »

```
root@grios-pc:/home/grios/uade_workplace/ssoo/tp1# exit
exit

~uade_workplace/ssoo/tp1
grios@personal-pc:~$ mkdir undir/subdir

~uade_workplace/ssoo/tp1
grios@personal-pc:~$ ls -l undir
total 4
drwxrwxr-x 2 grios grios 4096 ago 23 22:33 subdir
```

« Investigue como cambiar los permisos de manera recursiva sobre /undir para que todos sus archivos, subdirectorios y archivos dentro de los subdirectorios se vean afectados. »

Para cambiar permisos sobre un directorio de manera recursiva se debe ejecutar ’chmod’ con el flag ’-R’.

```
~uade_workplace/ssoo/tp1
grios@personal-pc:~$ ls -l
total 200
-rw-rw-r-- 1 grios grios 194796 ago 23 02:48 enunciado.pdf
drwxrwxr-x 3 grios grios   4096 ago 23 22:37 Informe
drwxr-xrwx 3 root  root    4096 ago 23 22:33 undir

~uade_workplace/ssoo/tp1
grios@personal-pc:~$ ls -l undir
total 4
drwxrwxr-x 2 grios grios 4096 ago 23 22:33 subdir

~uade_workplace/ssoo/tp1
grios@personal-pc:~$ chmod -R o+wrx undir
chmod: changing permissions of ’undir’: Operation not permitted

grios@personal-pc:~$ ls -l
total 200
-rw-rw-r-- 1 grios grios 194796 ago 23 02:48 enunciado.pdf
drwxrwxr-x 3 grios grios   4096 ago 23 22:37 Informe
drwxr-xrwx 3 root  root    4096 ago 23 22:33 undir

~uade_workplace/ssoo/tp1
grios@personal-pc:~$ ls -l undir
total 4
drwxrwxrwx 2 grios grios 4096 ago 23 22:33 subdir
```

7. Directorios

« ¿Para qué se usa el comando cd? »

Si bien usamos este comando durante casi todo el TP, tratamos de responder esta parte del TP ejecutando el siguiente comando sin exito:

```
~/uade_workplace/ssoo/tp1
grios@personal-pc:~$ man cd
No manual entry for cd
```

En base a nuestra experiencia, podemos decir que el comando cd (change directory) sirve para cambiar de directorios dentro una sesión del terminal

« Ejecute las siguientes variantes de cd y observe cuál fue el resultado obtenido: »

- textbfcd /

```
~
grios@personal-pc:~$ cd /

/
grios@personal-pc:~$ pwd
/
```

  Nos lleva al directorio root del sistema.

- textbfcd

```
/
grios@personal-pc:~$ cd

~
grios@personal-pc:~$ pwd
/home/grios
```

  Nos dirige al directorio root del usuario,

- textbfcd /etc

```
~
grios@personal-pc:~$ cd /etc

/etc
grios@personal-pc:~$ pwd
/etc
```

  Nos dirige al directorio root llamado '/etc'. En este directorio se persisten archivos de configuración del sistema operativo y paquetes instalados.

- textbfcd .

```
/etc
grios@personal-pc:~$ cd .

/etc
grios@personal-pc:~$ pwd
/etc
```

  Nos direcciona al directorio actual.

- textbfcd ..

```
/etc
grios@personal-pc:~$ cd ..

/
grios@personal-pc:~$ pwd
/
```

  Nos redireccionó al directorio superior, es decir, directorio root del sistema.

« Investigue que hacen los comandos: »

- textbfmkdir Este comando nos permite crear un directorio vacío.
- textbfrmdir Este comando nos permite borrar un directorio vacío.
- textbfrm Este comando nos permite borrar archivos, por defaul no borra directorios pero podría hacerlo si se le setea el flag '-r'

« Borre un directorio que no se encuentra vacío. »

```
~/uade_workplace/ssoo/tp1
grios@personal-pc:~$ mkdir temporal_dir

~/uade_workplace/ssoo/tp1
grios@personal-pc:~$ ls temporal_dir

~/uade_workplace/ssoo/tp1
grios@personal-pc:~$ touch temporal_dir/temp_file

~/uade_workplace/ssoo/tp1
grios@personal-pc:~$ rm -r temporal_dir
```

« Borre un directorio que no se encuentra vacío. »

```
~/uade_workplace/ssoo/tp1
grios@personal-pc:~$ cd

~
grios@personal-pc:~$ mkdir undir

~
grios@personal-pc:~$ cd undir
```

« Ingrese a dicho directorio, y tipee lo siguiente para crear muchos archivos “while (true) do ps >$RANDOM.text; done;”. Tipee ctrl.+c luego de 5 seg para finalizar el comando. Luego tipee ls para corroborar la creación de los archivos. »

Un detalle en este punto, simplemente notamos que el comando solicitado no genera los N archivos esperados, sino que se generaba uno solo. Supusimos que esto es porque la función definida dentro de $RANDOM devuelve un valor distinto cada vez que es consultado por el comando 'echo $RANDOM', como en la instrucción propuesta sólo se usa para persistir un file sin llamar al comando 'echo', el valor que devuelve $RANDOM nunca cambia haciendo que siempre se genere un solo archivo.

Dicho esto, cambiarmos el comando propuesto para que se ejecute un 'echo $RANDOM' luego de persistido el archivo.

```
~/undir
grios@personal-pc:~$ while (true) do ps > $RANDOM.txt; echo $RANDOM; done;

~
grios@personal-pc:~$ ls
10054.txt  15581.txt  1981.txt   24335.txt  30462.txt  5262.txt
10422.txt  15784.txt  19847.txt  24355.txt  30603.txt  5319.txt
10623.txt  15834.txt  19848.txt  2452.txt   30675.txt  5322.txt
10624.txt  15995.txt  20074.txt  24673.txt  30903.txt  5489.txt
10635.txt  15997.txt  20197.txt  24884.txt  3101.txt   5541.txt
10805.txt  16047.txt  20308.txt  25109.txt  31260.txt  5675.txt
10829.txt  16049.txt  20348.txt  25166.txt  31407.txt  572.txt
10982.txt  16081.txt  2037.txt   25269.txt  31550.txt  5775.txt
11000.txt  16094.txt  20386.txt  25333.txt  31599.txt  5934.txt
11582.txt  16335.txt  20388.txt  25562.txt  31631.txt  5978.txt
11632.txt  16398.txt  20663.txt  25625.txt  31781.txt  6138.txt
12090.txt  16446.txt  20782.txt  25775.txt  31819.txt  6372.txt
12106.txt  1654.txt   20877.txt  2584.txt   31839.txt  6437.txt
12120.txt  16668.txt  21009.txt  26048.txt  31856.txt  6476.txt
12139.txt  16743.txt  21023.txt  26061.txt  31934.txt  6506.txt
12237.txt  16871.txt  21321.txt  26135.txt  31940.txt  6525.txt
12465.txt  16944.txt  21527.txt  26330.txt  31941.txt  6544.txt
12530.txt  17020.txt  2156.txt   26343.txt  31965.txt  658.txt
12689.txt  17257.txt  21634.txt  2641.txt   32057.txt  6632.txt
12728.txt  17324.txt  21863.txt  26527.txt  32088.txt  6703.txt
12813.txt  17425.txt  21919.txt  26704.txt  32288.txt  6890.txt
12880.txt  17618.txt  21935.txt  26719.txt  32539.txt  6937.txt
12932.txt  17631.txt  21956.txt  26768.txt  32555.txt  7029.txt
13002.txt  17648.txt  21962.txt  2685.txt   32649.txt  7343.txt
13176.txt  17676.txt  22065.txt  26909.txt  32667.txt  7427.txt
13214.txt  1774.txt   2247.txt   27087.txt  32717.txt  7487.txt
13261.txt  17811.txt  22538.txt  27137.txt  32766.txt  7654.txt
13514.txt  18180.txt  22549.txt  27153.txt  3287.txt   7793.txt
13716.txt  18183.txt  22639.txt  27273.txt  3321.txt   8147.txt
13879.txt  1821.txt   22879.txt  27994.txt
```

« Tipee “rm *” e investigue que pasó. »

```
~/uade_workplace/ssoo/tp1
grios@personal-pc:~$ rm *
zsh: sure you want to delete more than 100 files in /home/grios/undir [yn]? y

grios@personal-pc:~$ ls
```

El comando rm (remove) borró todos los archivos de la carpeta actual. Esto se dió porque se hizo uso del wild card del asterisco (*), este comodín permite hacer referencia a todos los elemento del directorio actual y es por eso que se borraron.

8. Filtros

9. Redireccionamiento de E/S

10. Pipelines

« El carácter | (pipe) se usa para conectar la salida estándar de un comando con la entrada estándar de otro. Investíguelo tipeando “man bash” y llegando luego a la sección “Pipelines” (o “tuberías” en castellano). »

« Haciendo uso de ps y grep, liste todos los procesos del usuario root. »

```
~uade_workplace/ssoo/tp1
grios@personal-pc:~$ ps -fea | grep 'root'
root           1       0  0 ago22 ?        00:01:02 /sbin/init splash
root           2       0  0 ago22 ?        00:00:00 [kthreadd]
root           3       2  0 ago22 ?        00:00:00 [rcu_gp]
root           4       2  0 ago22 ?        00:00:00 [rcu_par_gp]
root           8       2  0 ago22 ?        00:00:00 [mm_percpu_wq]
root           9       2  0 ago22 ?        00:00:01 [ksoftirqd/0]
root          10       2  0 ago22 ?        00:00:46 [rcu_sched]
root          11       2  0 ago22 ?        00:00:00 [migration/0]
root          12       2  0 ago22 ?        00:00:00 [idle_inject/0]
root          14       2  0 ago22 ?        00:00:00 [cpuhp/0]
root          15       2  0 ago22 ?        00:00:00 [cpuhp/1]
root          16       2  0 ago22 ?        00:00:00 [idle_inject/1]
root          17       2  0 ago22 ?        00:00:00 [migration/1]
root          18       2  0 ago22 ?        00:00:00 [ksoftirqd/1]
root          21       2  0 ago22 ?        00:00:00 [cpuhp/2]
root          22       2  0 ago22 ?        00:00:00 [idle_inject/2]
root          23       2  0 ago22 ?        00:00:00 [migration/2]
root          24       2  0 ago22 ?        00:00:03 [ksoftirqd/2]
root          27       2  0 ago22 ?        00:00:00 [cpuhp/3]
root          28       2  0 ago22 ?        00:00:00 [idle_inject/3]
root          29       2  0 ago22 ?        00:00:00 [migration/3]
root          30       2  0 ago22 ?        00:00:14 [ksoftirqd/3]
root          33       2  0 ago22 ?        00:00:00 [kdevtmpfs]
root          34       2  0 ago22 ?        00:00:00 [netns]
root          35       2  0 ago22 ?        00:00:00 [rcu_tasks_kthre]
root          36       2  0 ago22 ?        00:00:00 [kauditd]
root          37       2  0 ago22 ?        00:00:00 [khungtaskd]
root          38       2  0 ago22 ?        00:00:00 [oom_reaper]
root          39       2  0 ago22 ?        00:00:00 [writeback]
root          40       2  0 ago22 ?        00:00:00 [kcompactd0]
root          41       2  0 ago22 ?        00:00:00 [ksmd]
root          42       2  0 ago22 ?        00:00:00 [khugepaged]
root         135       2  0 ago22 ?        00:00:00 [kintegrityd]
root         136       2  0 ago22 ?        00:00:00 [kblockd]
root         137       2  0 ago22 ?        00:00:00 [blkcg_punt_bio]
root         138       2  0 ago22 ?        00:00:00 [tpm_dev_wq]
root         139       2  0 ago22 ?        00:00:00 [ata_sff]
root         140       2  0 ago22 ?        00:00:00 [md]
root         141       2  0 ago22 ?        00:00:00 [edac-poller]
root         142       2  0 ago22 ?        00:00:00 [devfreq_wq]
root         144       2  0 ago22 ?        00:00:00 [watchdogd]
root         147       2  0 ago22 ?        00:00:19 [kswapd0]
root         148       2  0 ago22 ?        00:00:00 [ecryptfs-kthrea]
root         151       2  0 ago22 ?        00:00:00 [kthrotld]
root         152       2  0 ago22 ?        00:00:00 [acpi_thermal_pm]
root         154       2  0 ago22 ?        00:00:00 [vfio-irqfd-clea]
root         156       2  0 ago22 ?        00:00:00 [ipv6_addrconf]
root         167       2  0 ago22 ?        00:00:00 [kstrp]
root         171       2  0 ago22 ?        00:00:00 [kworker/u9:0-hci0]
root         187       2  0 ago22 ?        00:00:00 [charger_manager]
root         242       2  0 ago22 ?        00:00:00 [scsi_eh_0]
root         243       2  0 ago22 ?        00:00:00 [scsi_tmf_0]
root         244       2  0 ago22 ?        00:00:00 [scsi_eh_1]
root         245       2  0 ago22 ?        00:00:00 [scsi_tmf_1]
root         246       2  0 ago22 ?        00:00:00 [scsi_eh_2]
root         247       2  0 ago22 ?        00:00:00 [scsi_tmf_2]
root         248       2  0 ago22 ?        00:00:00 [scsi_eh_3]
root         249       2  0 ago22 ?        00:00:00 [scsi_tmf_3]
root         250       2  0 ago22 ?        00:00:00 [scsi_eh_4]
root         251       2  0 ago22 ?        00:00:00 [scsi_tmf_4]
root         252       2  0 ago22 ?        00:00:00 [scsi_eh_5]
root         253       2  0 ago22 ?        00:00:00 [scsi_tmf_5]
root         296       2  0 ago22 ?        00:00:00 [jbd2/sda6-8]
root         297       2  0 ago22 ?        00:00:00 [ext4-rsv-conver]
root         337       1  0 ago22 ?        00:00:03 /lib/systemd/systemd-journald
root         368       2  0 ago22 ?        00:00:00 [loop0]
root         371       2  0 ago22 ?        00:00:00 [loop1]
root         374       2  0 ago22 ?        00:00:00 [loop2]
root         379       2  0 ago22 ?        00:00:00 [loop3]
root         380       2  0 ago22 ?        00:00:00 [loop4]
root         381       2  0 ago22 ?        00:00:00 [loop5]
root         382       2  0 ago22 ?        00:00:00 [loop6]
root         383       2  0 ago22 ?        00:00:00 [loop7]
root         384       2  0 ago22 ?        00:00:00 [loop8]
root         385       2  0 ago22 ?        00:00:00 [loop9]
root         386       2  0 ago22 ?        00:00:00 [loop10]
root         388       1  0 ago22 ?        00:00:04 /lib/systemd/systemd-udevd
root         389       2  0 ago22 ?        00:00:00 [loop11]
root         482       2  0 ago22 ?        00:00:00 [cfg80211]
root         550       2  0 ago22 ?        00:00:00 [wl_event_handle]
```

```
root         554       2  0 ago22 ?        00:00:00 [cryptd]
root         572       2  0 ago22 ?        00:00:00 [kmemstick]
root         586       2  0 ago22 ?        00:00:00 [kworker/u9:2-hci0]
root         733       2  0 ago22 ?        00:00:04 [jbd2/sda7-8]
root         734       2  0 ago22 ?        00:00:00 [ext4-rsv-conver]
root         786       1  0 ago22 ?        00:00:01 /usr/lib/accountsservice/accounts-daemon
root         787       1  0 ago22 ?        00:00:13 /usr/sbin/acpid
root         791       1  0 ago22 ?        00:00:00 /usr/lib/bluetooth/bluetoothd
root         792       1  0 ago22 ?        00:00:00 /usr/sbin/cron -f
root         795       1  0 ago22 ?        00:00:13 /usr/sbin/NetworkManager --no-daemon
root         802       1  0 ago22 ?        00:00:02 /usr/sbin/irqbalance --foreground
root         803       1  0 ago22 ?        00:00:00 /usr/bin/python3 /usr/bin/networkd-dispatcher --run-startup-tri
root         807       1  0 ago22 ?        00:00:03 /usr/lib/policykit-1/polkitd --no-debug
root         810       1  0 ago22 ?        00:00:08 /usr/lib/snapd/snapd
root         811       1  0 ago22 ?        00:00:00 /usr/libexec/switcheroo-control
root         833       1  0 ago22 ?        00:00:03 /lib/systemd/systemd-logind
root         834       1  0 ago22 ?        00:00:02 /usr/sbin/thermald --no-daemon --dbus-enable
root         835       1  0 ago22 ?        00:00:07 /usr/lib/udisks2/udisksd
root         836       1  0 ago22 ?        00:00:00 /sbin/wpa_supplicant -u -s -O /run/wpa_supplicant
root         900       1  0 ago22 ?        00:00:00 /usr/sbin/ModemManager --filter-policy=strict
root         923       1  0 ago22 ?        00:00:00 /usr/bin/python3 /usr/share/unattended-upgrades/unattended-upgr
root         929       1  0 ago22 ?        00:00:07 /usr/bin/containerd
root         961       1  0 ago22 ?        00:00:00 /usr/sbin/gdm3
root         980       1  0 ago22 ?        00:00:02 /usr/sbin/apache2 -k start
root        1104       1  0 ago22 ?        00:00:11 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/containe
root        1139       2  0 ago22 ?        00:00:00 [iprt-VBoxWQueue]
root        1140       2  0 ago22 ?        00:00:00 [iprt-VBoxTscThr]
root        1257       1  0 ago22 ?        00:00:02 /usr/lib/upower/upowerd
root        1359       2  0 ago22 ?        00:00:00 bpfilter_umh
root        1772     961  0 ago22 ?        00:00:00 gdm-session-worker [pam/gdm-password]
root        1877       2  0 ago22 ?        00:00:00 [krfcommd]
root        5007       2  0 ago22 ?        00:00:02 [kworker/3:2H-events_highpri]
root        6197       1  0 00:00 ?        00:00:00 /usr/sbin/cupsd -l
root        6199       1  0 00:00 ?        00:00:00 /usr/sbin/cups-browsed
root       22688       2  0 03:36 ?        00:00:01 [kworker/0:2H-events_highpri]
root       30651       2  0 18:48 ?        00:00:00 [irq/30-mei_me]
root       43634       2  0 20:09 ?        00:00:01 [kworker/1:1H-events_highpri]
root       44977       2  0 20:28 ?        00:00:01 [kworker/0:0H-events_highpri]
root       49773       2  0 21:29 ?        00:00:02 [kworker/0:1-events]
root       49832       2  0 21:33 ?        00:00:00 [kworker/1:2-events]
root       49833       2  0 21:33 ?        00:00:01 [kworker/1:3-events]
root       49894       2  0 21:41 ?        00:00:06 [kworker/u8:0+events_unbound]
root       50022       2  0 22:01 ?        00:00:03 [kworker/2:0-events]
root       50057       2  0 22:02 ?        00:00:01 [kworker/3:1H-events_highpri]
root       50197       2  0 22:03 ?        00:00:00 [kworker/0:2]
root       52077       2  0 22:34 ?        00:00:01 [kworker/3:0-events]
root       52786       2  0 22:41 ?        00:00:00 [kworker/3:1-events]
root       52868       2  0 22:42 ?        00:00:00 [kworker/2:1-events]
root       53321       2  0 22:46 ?        00:00:00 [kworker/1:0H-events_highpri]
root       53655       2  0 22:48 ?        00:00:00 [kworker/2:2H-events_highpri]
root       53698       2  0 22:48 ?        00:00:02 [kworker/u8:2-events_unbound]
root       54199       2  0 22:55 ?        00:00:00 [kworker/2:0H-events_highpri]
root       54473       2  0 23:01 ?        00:00:02 [kworker/u8:1-events_unbound]
root       55262       2  0 23:10 ?        00:00:01 [kworker/u8:4+events_unbound]
root       56134       2  0 23:23 ?        00:00:00 [kworker/u8:3-events_unbound]
root       56536       2  0 23:25 ?        00:00:00 [kworker/2:2-events]
```

11. Vim

12. Shell scripting y otras cuestiones