

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 simbólicos, 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 cuentas de superusuario o root son cuentas que poseen permisos de administrador, es decir, tienen privilegios y permisos para realizar acciones sobre el sistema. Contar con estos permisos permitirá ejecutar instrucciones sumamente útiles 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 loggarse, 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 necesario 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 táctica que antes podemos 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 {source_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 diferencia que se usa un protocolo ssh estableciendo un canal seguro ya que la 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~$ ls /var | grep 'bashrc'
-rw-r--r-- 1 root root 3,1K ago 23 12:55 .bashrc

~uade_workplace/ssoo/tp1
grios@personal-pc~$ ls /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 permisos 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 mencionados 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 propietario 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 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 permisos de un directorio se estructuran con 10 caracteres (10 bits), los cuales se ven representados de la siguiente forma:

	6	5	4	3	2	1	0
	d	r	w	X	r	w	X
directorio	permiso del usuario			permisos de grupo			permisos para otros

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

« Loguése 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+rx 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+wx 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

8. Filtros

9. Redireccionamiento de E/S

10. Pipelines

11. Vim

12. Shell scripting y otras cuestiones