

Práctica 1: Uso del S.O. GNU/Línx- GRUPAL (Puntuable)

1. Objetivos

- Que se familiarice con el uso de un Sistema Operativo distinto al común denominador
- Que disponga de un ambiente de ejecución multitarea y multiprogramado para realizar las prácticas obligatorias y conozca los principales comandos de GNU/Linux.
- Que se compenetre de la normativa en la presentación de informes o trabajos prácticos

2. Especificación o Descripción:

- El presente trabajo práctico está pensado para que sea confeccionado en una máquina (preferentemente de su propiedad), con GNU/Linux instalado (la instalación no es parte del presente trabajo práctico). El mismo puede ser realizado en una máquina virtual o desde la ejecución de los llamados live cd (ejecutar el sistema operativo desde un cd/dvd o un pendrive). En la sección de recursos encontrarán links de descargas y tutoriales en youtube que si bien son los recomendados por la cátedra no es obligatorio usar la misma distro de Linux ni la misma máquina virtual.
- Se deberá confeccionar una monografía sobre el cuestionario propuesto por la cátedra y basado en el uso del sistema operativo GNU/Linux. Tener en cuenta que la **monografía** no es una novela, por lo cual debe cumplir con todos los requisitos y tener todas las secciones necesarias para ser considerada una **monografía**, Las secciones mínimas son:

- Carátula
- Índice
- Objetivos del trabajo
- Introducción o plan de trabajo
- Cuerpo (Desarrollo)
- Conclusiones
- Referencias Bibliográficas (Si las hay)
- Anexos.

Se deberá confeccionar la **monografía** respondiendo el cuestionario vertiendo las dificultades encontradas.

3. Requisitos

- **Formato de entrega:** La monografía se entregará en formato de Microsoft Word.
- **Fecha de entrega:** **de acuerdo a planificación**
- **Observación:** La entrega es obligatoria para esta práctica y la aprobación tanto de la monografía como del cuestionario es requisito para la regularización de la materia.

4. Recursos

Links de descarga y tutoriales.

Distro de Linux recomendada:

-[Ubuntu 20.04 64bits.](#)

Software para crear usb booteable:

-<https://unetbootin.github.io/>

-Tutorial: <https://www.youtube.com/watch?v=ZpLVpdN9gTE>

Software de máquina virtual:

-[Máquina virtual virtualbox.](#)

-Tutorial: <https://www.youtube.com/watch?v=GEx046EHphI>

5. Cuestionario.

- **Preguntas:** A continuación, se detallan todas las preguntas y ejercicios que deberán ser resueltos. Tenga en cuenta que salvo en los momentos que indica que debe estar sesionado como root, en el resto de los ejercicios debe estar conectado como usuario común.

1. INTRODUCCIÓN

- 1.1. ¿Qué es la cuenta de superusuario (root) y para qué se utiliza?
- 1.2. Ingresar al sistema como superusuario (root), y realizar los siguientes pasos:
 - 1.3. adduser <apellido> (reemplazar <apellido> por el suyo).
 - 1.4. passwd <apellido> (Ingrese una contraseña (password) a su elección).
 - 1.5. logout
- 1.6. Indique claramente qué efectuaron estos comandos, e indique qué archivo/s fueron modificados (Dentro del directorio /etc).
- 1.7. ¿En qué directorio se encuentran los comandos utilizados en los puntos 1.3 y 1.4?

2. TECLADO / TERMINALES

- 2.1. ¿Qué sucede si tecleo cat /e <tab> p <tab>? (donde tab es la tecla tabulación). Presione <tab> nuevamente ¿Qué pasó ahora?
- 2.2. ¿Qué sucede si tecleo cat /e <tab> pas <tab>?
- 2.3. En este punto analizaremos las distintas terminales que hay en un sistema GNU/Linux. Ejecute los siguientes comandos e indique cuál fue el resultado:
 - 2.3.1. who
 - 2.3.2. Presione las teclas <ctrl> y <alt>, y sin soltarlas presione cualquiera de las teclas de función (de F3 a F6 para Ubuntu). En la pantalla debería aparecer el login del sistema, de lo contrario, ejecute el paso nuevamente presionando otra tecla de función. Si ya tiene el login del sistema vuelva a conectarse.

- 2.3.3. Ejecute nuevamente el comando who. ¿Qué diferencias encuentra con la primera vez que lo ejecutó?
- 2.3.4. Ejecute el comando who am i ¿qué muestra?, ¿Qué diferencias tiene con el comando ejecutado en el punto anterior?
- 2.3.5. Repita el paso 2.3.2 y el 2.3.3 hasta que no encuentre ninguna sesión para abrir.
- 2.3.6. Una vez terminado el punto anterior, Ud. se encontrará sesionado en el sistema como mínimo cinco veces. Lo que acaba de hacer es abrir cuatro terminales virtuales (que podrían ser usadas por distintos usuarios, con diferentes perfiles), en la misma máquina. Así como existen terminales virtuales dentro del mismo equipo, si Ud. cuenta con una red, o con terminales tipo serie, podría abrir tantas sesiones de trabajo como Ud. quiera o necesite. Investigue e indique cómo se denominan los distintos tipos de terminales, y cuáles son los archivos que las representan (tip: busque en el directorio /dev).
- 2.3.7. Use el comando logout para cerrar las sesiones abiertas en todas las terminales virtuales y vuelva a la sesión gráfica (<ctrl> <alt> F2 para Ubuntu).

3. AYUDA

3.1. INFO: Info es un programa para leer documentación. Este se compone de una estructura del tipo árbol, dividido en nodos de información. Cada nodo describe un específico tópico con un determinado nivel de detalle.

3.1.1. Ingrese a info y responda:

- 3.1.1.1. ¿Cómo se llama el nodo raíz de Info?
 - 3.1.1.2. Ubique el cursor en la línea (* cp:) y presione ENTER.
 - 3.1.1.3. ¿Qué sucedió?
 - 3.1.1.4. ¿Cómo se llama este nodo?
 - 3.1.1.5. ¿Cuál es el próximo nodo?
 - 3.1.1.6. ¿Cómo puedo moverme al próximo nodo?
 - 3.1.1.7. ¿Cómo puedo moverme al nodo anterior?
- 3.1.2. Presione la tecla 'u'.
- 3.1.2.1. ¿Qué sucedió?
 - 3.1.2.2. ¿En qué nodo se encuentra?
- 3.1.3. Repita el punto 3.1.2. hasta que llegue a la raíz de Info.
- 3.1.3.1. ¿Con qué tecla puedo volver directamente a este nodo?.
 - 3.1.3.2. ¿Cuál es el método directo para acceder al nodo cp? (tip: sin desplazar el cursor).

3.1.4. ¿Cómo puedo buscar una palabra clave dentro de un nodo?

3.1.5. ¿Cómo puedo buscar la siguiente palabra clave, buscada anteriormente?

3.1.6. ¿Cómo puedo salir de Info? -hágalo.

3.2. MAN: man es un programa que formatea y muestra las páginas del manual.

3.2.1. ¿Cuál es la diferencia entre man e info?

3.2.2. ¿Cómo puedo ver la información de un determinado comando?

3.2.3. ¿Cómo puedo buscar una palabra clave dentro de la página del manual?

3.2.4. ¿Cómo puedo salir?

3.2.5. ¿Cómo hago para buscar una palabra clave determinada en todas las páginas del manual?

3.2.6. ¿Qué es lo sucede al realizar lo siguiente?

3.2.6.1. man

3.2.6.2. man man

- 3.2.6.3. man cp
- 3.2.6.4. man printf
- 3.2.6.5. man fprintf
- 3.2.6.6. man sprintf
- 3.2.6.7. man 3 printf

3.2.7. Si los comandos del punto anterior le dieron un mensaje de error, instale el paquete manpages-dev y vuelva a ejecutarlos (utilice sudo apt install manpages-dev si está usando Ubuntu).

3.2.8. Del punto anterior, responder:

3.2.8.1. Al invocar man junto con fprintf y sprintf muestra la misma página. ¿Porqué no muestra la misma página al invocarlo con printf?. (TIP: vea el punto 3.2.6.2).

3.2.8.2. ¿Cómo puedo invocar al man para ver directamente la función printf del lenguaje C?.

3.3. HELP: help es la ayuda que ofrece el shell de GNU/LINUX para utilizar sus comandos.

3.3.1. ¿Cuál es la diferencia entre help e info?.

3.3.2. ¿Cuál es la diferencial entre help y man?.

3.3.3. ¿Qué sucede al invocar al help?.

3.3.4. ¿Cómo puedo ver la información de un determinado comando?

3.3.5. ¿Qué sucede al utilizar el comando help cd?

3.4. whereis

3.4.1. ¿Qué sucede al utilizar el comando whereis cp?

3.4.2. ¿Qué es la información que se muestra por pantalla al ejecutar el punto anterior?

3.4.3. ¿Qué ocurre si se ejecuta whereis * sobre un directorio? (Tip: si no pasa nada, intentelo nuevamente pero primero ejecute cd /bin)

3.4.4. ¿Cuál es la diferencia entre whereis y find?

3.5. whatis

3.5.1. ¿Qué sucede al utilizar el comando whatis cp?

3.5.2. Cambie al directorio /bin, y ejecute el comando whatis * ¿Qué ocurrió?

3.5.3. Utilice el comando apropos passwd y whatis passwd. Enumere las diferencias encontradas en el resultado de cada uno de los comandos.

4. DIRECTORIOS

4.1. ¿Para qué se usa el comando cd? Ejecute las siguientes variantes de cd e indique cuál fue el resultado obtenido:

4.1.1. cd /

4.1.2. cd

4.1.3. cd /etc

4.1.4. cd..

4.1.5. cd ..

4.2. Bash sobre directorios:

4.2.1. ¿Cuál/es son las diferencias entre el path absoluto y el path relativo?

4.2.2. ¿Qué es lo que realizan las siguientes operaciones? (tip: si no encuentra la diferencia primero haga cd /, y luego vuelva a intentar)

4.2.2.1. cd ~

4.2.2.2. cd -

4.2.3. ¿Cuál es la diferencia entre cd ~ y cd -?

- 4.3. Operaciones con directorios:
 - 4.3.1. ¿Con qué comando se puede crear un directorio?
 - 4.3.2. ¿Con qué comando se puede borrar un directorio?
 - 4.3.3. ¿Qué sucede si el directorio no está vacío?
 - 4.3.4. ¿Cómo puedo salvar la situación anterior? (Sin borrar uno a uno los archivos existentes).
- 4.4. ¿Qué significa la expresión ./ cuando se utiliza delante de un archivo? ¿Para qué sirve?
- 4.5. ¿Cómo puede moverse entre directorios sin utilizar el PATH completo?

5. ARCHIVOS

- 5.1. ¿Qué hacen los siguientes comandos?
 - 5.1.1. cp
 - 5.1.2. mv
 - 5.1.3. rm
 - 5.1.4. rcp
 - 5.1.5. rsh
 - 5.1.6. scp
 - 5.1.7. ssh
- 5.2. ¿Con qué comando puedo concatenar el contenido de dos archivos?
- 5.2.1. ¿Se puede usar ese comando para otra cosa?
- 5.3. Haga un ls -l /dev
 - 5.3.1. ¿Qué significa el primer carácter?
 - 5.3.2. ¿Cuáles son todos los posibles valores que puede contener ese campo y que significa cada uno?
- 5.4. ¿Para qué sirve el comando touch? ¿qué utilidad le encuentra?

6. PERMISOS

- 6.1. Teniendo en cuenta el ls -l anterior, ¿indique que son los siguientes 9 caracteres? (sin considerar el primero sobre el que ya respondió anteriormente)
- 6.2. ¿qué significan cada caracter? ¿cómo están agrupados?
- 6.3. ¿Cómo se asignan los permisos? (detalle los comandos).
- 6.4. ¿Qué son el owner, y el group de un archivo? ¿Se pueden cambiar?
- 6.5. Intente cambiar los permisos de un archivo perteneciente al root (sesionado como usuario). Explique qué sucedió.
- 6.6. Explique la forma de cambiar los permisos con valores en octal.
- 6.7. ¿Cuál es el significado de los permisos en los directorios (se debe indicar que indica una r, una w, y una x)?

7. FILTROS

- 7.1. ¿Cuál es la diferencia de los comandos more, less y cat? De un ejemplo de cada uno.
- 7.2. ¿Cuál es la diferencia entre tail y head?
- 7.3. ¿Qué es lo que realiza el comando uniq?
- 7.4. ¿Qué es lo que realiza el comando grep?

8. VI

8.1. Ejecute la siguiente instrucción: vi \$HOME/prueba.txt ¿Qué sucedió? Ahora ejecute todos los pasos detallados a continuación.

8.1.1. Escriba la siguiente frase: "Este es el archivo prueba.txt de <nombre y apellido>"

8.1.2. ¿Qué tuvo que hacer para poder escribir la frase?

8.1.3. Guarde el archivo, y salga del editor. ¿Qué comando utilizó?

8.1.4. Ingrese nuevamente al archivo.

8.1.5. Incorpore al inicio del archivo el siguiente párrafo (los acentos pueden ser evitados):
"Sistemas Operativos

Curso de los días <día de cursada>

Trabajo Práctico 1

Alumno: <su nombre y apellido aquí>

Legajo: <su Legajo aquí>

Documento: <su documento aquí>"

8.1.6. Describa todos los pasos que tuvo que realizar.

8.1.7. Guarde el archivo y continúe la edición. ¿Qué comandos utilizó?

8.1.8. Borre la línea de "Legajo". Indique por lo menos dos formas de realizarlo.

8.1.9. Invierta el orden de las líneas "Curso y TP". No está permitido rescribirlas. ¿Qué comandos utilizó?

8.1.10. Ubíquese en la línea 2 (dos) del archivo. No está permitido usar las teclas del cursor, ni el mouse. ¿Qué comando utilizó?

8.1.11. Marque para copiar las líneas 2, 3, y 4 (todas juntas, no de a una a la vez). ¿Cómo lo realizó?

8.1.12. Ubíquese al final del archivo (sin usar las teclas del cursor), y pegue dos veces el contenido del buffer. ¿Qué comando usó?

8.1.13. Deshaga uno de los copiados. No está permitido borrar línea por línea, ni carácter a carácter. ¿Qué comando usó?

8.1.14. ¿Cómo busco la palabra "Documento"? ¿Cómo busco la segunda ocurrencia de una palabra?

8.1.15. ¿Cómo puedo reemplazar la palabra "Documento" por "Documento:" (sin borrar, o realizar el reemplazo a mano)?

8.1.16. Guarde el archivo y salga.

8.1.17. Ejecutar "vi miprog.c"

introducir el texto:

```
principal ()  
{  
printf (" 1/3/2001 ----- listo");  
printf (" 1/3/2002 ----- listo");  
printf (" 1/3/2003 ----- listo");  
printf (" 1/3/2004 ----- listo");  
printf (" 1/3/2005 ----- listo");  
printf (" 1/3/2006 ----- No listo");  
}
```

Luego presionar :x

ahora ejecute los siguientes comandos:

gcc miprog.c -o miprimerprog (si no tiene instalado gcc, hágalo usando el comando sudo apt install gcc).

./miprimerprog

¿Cuál es la salida?

volver a ejecutar vi miprog.c

poner “:%s/V3/VMarzo/g” ¿Que paso al ejecutar esto?

Corrija los errores que surgieron al utilizar el comando gcc.

volver a ejecutar: x

ahora volver a ejecutar:

gcc miprog.c -o miprimerprog

./miprimerprog

¿Cual es la salida ahora?

9. VARIABLES DE ENTORNO

9.1. ¿Qué son las variables de entorno y para qué sirven?

9.1.1. Escriba el contenido y explique el significado de las siguientes variables: HOME / LOGNAME / PATH / HOSTNAME / IFS

9.1.2. ¿Qué comando usó para ver el contenido de las variables del punto anterior?

9.1.3. Cree una variable de entorno HOLA que contenga el mensaje “Hola mundo”.

9.1.4. ¿Cuál es el uso que le da el sistema a la variable PATH? ¿Qué ocurre si intenta ejecutar un comando que no se encuentra ubicado en alguno de los directorios que contiene la variable? ¿Cómo lo soluciona?

9.1.5. ¿Por qué existen las variables PS1 y PS2? ¿Qué es un comando multilínea?

RESPUESTAS

1.INTRODUCCIÓN

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

El superusuario (root), se utiliza para tener todos los permisos en la consola de Linux y al tener todos los permisos se podrá utilizar las herramientas sin límites. Ya que siendo un usuario normal no tendrás todos los permisos de las herramientas o las tendrías limitadas.

//• Como volverte super usuario = <https://www.youtube.com/watch?v=J6Ah7Bqv09U>

el comando para volverte Root es = **sudo su || sudo -i** y una vez ingresado en la consola te pedirá que ingreses la contraseña y te volverás root

```
maxi@maxi-VirtualBox:~$ sudo su
[sudo] contraseña para maxi:
root@maxi-VirtualBox:/home/maxi#
```

1.2. Ingresar al sistema como superusuario (root), y realizar los siguientes pasos:

1.3. **adduser <apellido> (reemplazar <apellido> por el suyo).**

- **adduser** = te permite Crear un nuevo usuario (única vez sin poder modificarlo), desde el root, si no te tira un error.
- debes de ingresar tu nombre de usuario, pero si no quieres solo apretas enter.
- Ingresar una contraseña para ese usuario.
- y así con todos los datos que te va pidiendo.

//<https://www.youtube.com/watch?v=RyKIAAnSGdHM>

1.2. Ingresar al sistema como superusuario (root), y realizar los siguientes pasos:

1.4. **passwd <apellido> (Ingrese una contraseña (password) a su elección).**

- **passwd** = el comando para cambiar la contraseña, se puede utilizar para cambiar a sí mismo y para otros usuarios ya existentes.

1.2. Ingresar al sistema como superusuario (root), y realizar los siguientes pasos:

1.5. **logout**

- **logout** = se utiliza para cerrar la terminal, pero en linux ubuntu se reemplaza por el comando **exit**, ya que no se utiliza en linux el comando Logout.

y **logout** también se utiliza para salir de una cuenta iniciada dentro de la terminal

1.6. Indique claramente qué efectuaron estos comandos, e indique qué archivo/s fueron modificados (Dentro del directorio /etc).

los usuarios creados por **adduser** están en la dirección :

```
maxi@maxi-VirtualBox:~$ sudo su
[sudo] contraseña para maxi:
root@maxi-VirtualBox:/home/maxi# cd /home
root@maxi-VirtualBox:/home# ls
dani maxi maxi1
root@maxi-VirtualBox:/home#
```

y las contraseñas creadas en **passwd** están creadas en la dirección:

```
maxi@maxi-VirtualBox:~$ cd /etc
maxi@maxi-VirtualBox:/etc$ ls
acpt                  hdparm.conf      pmn2ppa.conf
adduser.conf          host.conf       polkit-1
alsa                  hostid          popularity-contest.conf
alternatives          hostname        ppp
anacrontab            hosts          profile
apg.conf              hosts.allow    profile.d
apm                  hosts.deny     protocols
apparmor              hp              pulse
apparmor.d            ifplugd        python3
apport                init           python3.8
appstream.conf        init.d         rc0.d
apt                  initramfs-tools rci.d
avahi                inputrc        rc2.d
bash.bashrc            inserv.conf.d rc3.d
bash_completion        iproute2       rc4.d
bash_completion.d     issue          rc5.d
bindresvport.blacklist issue.net     rc6.d
binfmt.d              kernel         rcS.d
bluetooth             kernel-img.conf resolv.conf
brlapi.key             kernelloops.conf rmt
brltty                ldap           rpc
brltty.conf            ld.so.cache   rsyslog.conf
ca-certificates        ld.so.conf    rsyslog.d
ca-certificates.conf  ld.so.conf.d  rygel.conf
ca-certificates.conf.d legal          sane.d
calendar              libao.conf    security
chatscripts            libaudit.conf selinux
console-setup          libblockdev  sensors3.conf
cracklib              libnl-3       sensors.d
cron.d                libpaper.d   services
cron.daily             libreoffice  sgml
cron.hourly            locale.alias shadow
cron.monthly           locale.gen   shadow-
cron.monthly           localtime    shells
cron.weekly            logcheck     skel
cups                  login.defs   snmp
cupshelpers            logrotate.conf speech-dispatcher
dbus-1                lsb-release  ssh
dconf                 logrotate.d  ssl
debconf.conf           ltrace.conf  subgid
debian_version         machine-id  subgid-
default               magic        subuid
```

```
deluser.conf          magic.mime           subuid-
depmod.d             mailcap              sudoers
dhcp                 mailcap.order        sudoers.d
dictionaries-common manpath.config      sysctl.conf
dpkg                 mime.types          sysctl.d
e2scrub.conf         mke2fs.conf         systemd
emacs                modprobe.d          terminfo
environment          modules              thermald
environment.d        modules-load.d    thunderbird
ethertypes           mtab                timezone
firefox              mtools.conf         tmpfiles.d
fonts                mysql               ubuntu-advantage
fprintd.conf         nanorc              ucf.conf
fstab                netplan              udev
fuse.conf            network              udisks2
fwupd                networkd-dispatcher update-manager
gai.conf              NetworkManager      update-motd.d
gamemode.ini         networks             update-notifier
gdb                  newt                UPower
gdm3                nsswitch.conf       usb_modeswitch.conf
geoclue              openvpn             usb_modeswitch.d
ghostscript          opt                vim
glvnd                os-release          vtrgb
gnome                PackageKit          vulkan
groff               pam.conf            wgetrc
group                pam.d               wpa_supplicant
group-               papersize          X11
grub.d               passwd              xattr.conf
gshadow              passwd-             xdg
gshadow-             pcmcia             xml
gss                  perl               zsh_command_not_found
gtk-2.0              pki                max@maxi-VirtualBox:/etc$ █
gtk-3.0
```

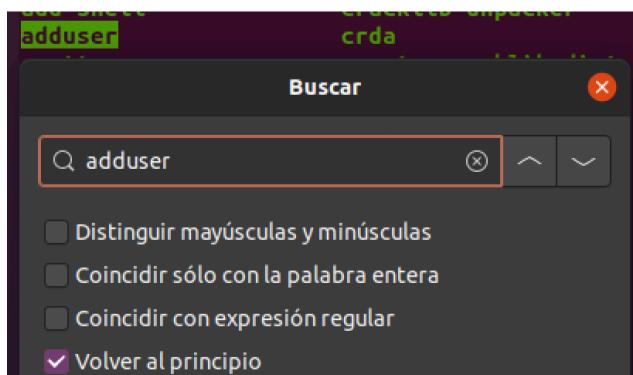
y para modificar la carpeta se utiliza el siguiente comando :

```
maxi@maxi-VirtualBox:/etc$ gedit passwd
```

1.7. ¿En qué **directorio** se encuentran los comandos utilizados en los puntos 1.3 y 1.4?

El directorio que se encuentra el **adduser** es en el :

```
maxi@maxi-VirtualBox:/usr/bin$ cd /usr/sbin  
maxi@maxi-VirtualBox:/usr/sbin$ ls
```



el directorio que se encuentra el **passwd** es en el :

```
maxi@maxi-VirtualBox:~$ cd /usr/bin  
maxi@maxi-VirtualBox:/usr/bin$ ls  
crown parecord  
chrt partx  
chsh passwd  
chvt paste  
cisco1 rootsize
```

2. TECLADO / TERMINALES

2.1. ¿Qué sucede si tecleo **cat /e <tab> p <tab>**? (donde **tab** es la tecla tabulación).

cat /e <tab> p <tab>

```
maxi@maxi-VirtualBox:~$ cat /etc/p
```

Presione <tab> nuevamente ¿Qué pasó ahora?

```
maxi@maxi-VirtualBox:~$ cat /etc/p
pam.conf          pki/                  profile
pam.d/            pm/                  profile.d/
papersize         pnm2ppa.conf        protocols
passwd            polkit-1/             pulse/
passwd-           popularity-contest.conf python3/
pcmcia/           ppp/                 python3.8/
perl/             printcap
maxi@maxi-VirtualBox:~$ cat /etc/p
```

2.2. ¿Qué sucede si tecleo `cat /e <tab> pas <tab>`?

si lo escribo haciendo cada uno de sus pasos

```
maxi@maxi-VirtualBox:~$ cat /etc/passwd?
```

Y me muestra esto:

```
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
systemd-network:x:100:102:systemd Network Management,,,:/run/systemd:/usr/sbin/nologin
systemd-resolve:x:101:103:systemd Resolver,,,:/run/systemd:/usr/sbin/nologin
systemd-timesync:x:102:104:systemd Time Synchronization,,,:/run/systemd:/usr/sbin/nologin
messagebus:x:103:106::/nonexistent:/usr/sbin/nologin
syslog:x:104:110::/home/syslog:/usr/sbin/nologin
_apt:x:105:65534::/nonexistent:/usr/sbin/nologin
tss:x:106:111:TPM software stack,,,:/var/lib/tpm:/bin/false
uuidd:x:107:114::/run/uuidd:/usr/sbin/nologin
tcpdump:x:108:115::/nonexistent:/usr/sbin/nologin
avahi-autoipd:x:109:116:Avahi autoip daemon,,,:/var/lib/avahi-autoipd:/usr/sbin/nologin
usbmux:x:110:46:usbmux daemon,,,:/var/lib/usbmux:/usr/sbin/nologin
rtkit:x:111:117:RealtimeKit,,,:/proc:/usr/sbin/nologin
dnsmasq:x:112:65534:dnsmasq,,,:/var/lib/misc:/usr/sbin/nologin
cups-pk-helper:x:113:120:user for cups-pk-helper service,,,:/home/cups-pk-helper:/usr/sbin/nologin
speech-dispatcher:x:114:29:Speech Dispatcher,,,:/run/speech-dispatcher:/bin/false
avahi:x:115:121:Avahi mDNS daemon,,,:/var/run/avahi-daemon:/usr/sbin/nologin
kernoops:x:116:65534:Kernel Oops Tracking Daemon,,,:/usr/sbin/nologin
saned:x:117:123::/var/lib/saned:/usr/sbin/nologin
nm-openvpn:x:118:124:NetworkManager OpenVPN,,,:/var/lib/openvpn/chroot:/usr/sbin/nologin
hplip:x:119:7:HPLIP system user,,,:/run/hplip:/bin/false
whoopsie:x:120:125::/nonexistent:/bin/false
colord:x:121:126:colord colour management daemon,,,:/var/lib/colord:/usr/sbin/nologin
geoclue:x:122:127::/var/lib/geoclue:/usr/sbin/nologin
pulse:x:123:128:PulseAudio daemon,,,:/var/run/pulse:/usr/sbin/nologin
gnome-initial-setup:x:124:65534::/run/gnome-initial-setup:/bin/false
```

```
gdm:x:125:130:Gnome Display Manager:/var/lib/gdm3:/bin/false
sssd:x:126:131:SSSD system user,,,:/var/lib/sssd:/usr/sbin/nologin
maxi:x:1000:1000:Maxi,,,:/home/maxi:/bin/bash
systemd-coredump:x:999:999:systemd Core Dumper:/:/usr/sbin/nologin
maxi1:x:1001:1001:maxi1,,,:/home/maxi1:/bin/bash
maxi2:x:1002:1002:,,,:/home/maxi2:/bin/bash
maxi@maxi-VirtualBox:~$
```

Explicación de lo sucedido:

describe información útil sobre las cuentas de usuario definidas en el sistema. Cada línea de la información del usuario está representada por 7 campos que están separados (delimitados) por un símbolo de dos puntos.

cat /etc/passwd:

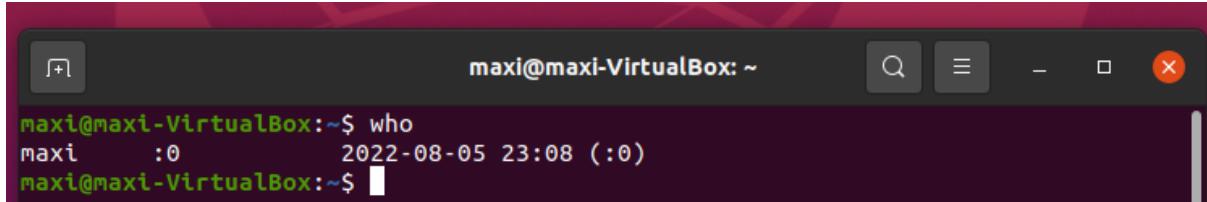
```
> cat /etc/passwd
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
↓ ↓ ↓ ↓ ↓ ↓ ↓
1 2 3 4 5 6 7
1: Username
2: Password (Encrypted)
3: User ID (UID)
4: Group ID (GID)
5: User ID Info (GECOS)
6: User Home Directory
7: User Login Shell
```

1. **Username:** una cadena única en una máquina que se utiliza para iniciar sesión en el sistema. La longitud máxima del nombre de usuario está restringida a 32 caracteres.
2. **Password:** Cifrada
3. **User ID (UID):** un identificador único para cada usuario, representado por un número.
4. **ID group(GID):** un identificador único para grupos de usuarios, representado por un número.
5. **User ID Info (GECOS):** valores separados por comas (nombre completo, número de habitación, número de teléfono del trabajo, número de teléfono de casa, otra información de contacto) que contienen información detallada sobre el usuario.
6. **User Home Directory:** la ruta absoluta al directorio de inicio del usuario. En la mayoría de las distribuciones de Linux, los directorios de inicio de los usuarios reciben el nombre de los nombres de usuario y se crean en el /homedirectorio.
7. **User Login Shell:** del usuario: la ruta absoluta al shell de inicio de sesión del usuario. En la mayoría de las distribuciones de Linux, el shell de inicio de sesión de usuario predeterminado es Bash.

<https://cybersophia.net/linux/etc-passwd-file-in-linux/#:~:text=To%20display%20the%20content%20of,in%20Linux%2C%20as%20described%20below.&text=To%20filter%20the%20output%20of,in%20Linux%2C%20as%20shown%20below>

2.3. En este punto analizaremos las distintas terminales que hay en un sistema GNU/Linux. Ejecute los siguientes comandos e indique cuál fue el resultado:

2.3.1. who



```
maxi@maxi-VirtualBox:~$ who
maxi      :0          2022-08-05 23:08 (:0)
maxi@maxi-VirtualBox:~$
```

2.3.2. Presione las teclas <ctrl> y <alt>, y sin soltarlas presione cualquiera de las teclas de función (de F3 a F6 para Ubuntu). En la pantalla debería aparecer el login del sistema, de lo contrario, ejecute el paso nuevamente presionando otra tecla de función. Si ya tiene el login del sistema vuelva a conectarse.



```
Ubuntu 20.04.4 LTS maxi-VirtualBox tty3

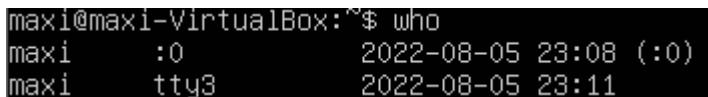
maxi-VirtualBox login: maxi
Password:
Welcome to Ubuntu 20.04.4 LTS (GNU/Linux 5.15.0-43-generic x86_64)

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

Se pueden aplicar 82 actualizaciones de forma inmediata.
Para ver estas actualizaciones adicionales ejecute: apt list --upgradable

Your Hardware Enablement Stack (HWE) is supported until April 2025.
Last login: Fri Aug  5 22:49:30 -03 2022 on tty4
maxi@maxi-VirtualBox:~$ _
```

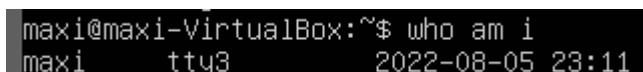
2.3.3. Ejecute nuevamente el comando **who**. ¿Qué diferencias encuentra con la primera vez que lo ejecutó?



```
maxi@maxi-VirtualBox:~$ who
maxi      :0          2022-08-05 23:08 (:0)
maxi      tty3        2022-08-05 23:11
```

La diferencia es que antes no aparecía **tty3** y ahora se muestra como un nuevo usuario.

2.3.4. Ejecute el comando **who am i** ¿qué muestra?, ¿Qué diferencias tiene con el comando ejecutado en el punto anterior?



```
maxi@maxi-VirtualBox:~$ who am i
maxi      tty3        2022-08-05 23:11
```

muestra el identificador del usuario actual.

2.3.5. Repita el paso 2.3.2 y el 2.3.3 hasta que no encuentre ninguna sesión para abrir.

```
maxi@maxi-VirtualBox:~$ who
maxi      :0          2022-08-05 23:08 (:0)
maxi      tty3        2022-08-05 23:11
maxi      tty4        2022-08-05 23:17
maxi      tty5        2022-08-05 23:18
maxi      tty6        2022-08-05 23:18
maxi@maxi-VirtualBox:~$
```

2.3.6. Una vez terminado el punto anterior, Ud. se encontrará sesionado en el sistema como mínimo cinco veces. Lo que acaba de hacer es abrir cuatro **terminales virtuales** (que podrían ser usadas por distintos usuarios, con diferentes perfiles), en la misma máquina. Así como existen terminales virtuales dentro del mismo equipo, si Ud. cuenta con una red, o con terminales tipo serie, podría abrir tantas sesiones de trabajo como Ud. quiera o necesite.
Investigue e indique cómo se denominan los distintos tipos de terminales,

Terminal tty en GNU/Linux

La **tty** es la consola que nos permite en GNU/Linux acceder a nuestro sistema operativo fuera de su entorno gráfico (X-Window). En general disponemos de hasta seis terminales. Para acceder:

Para acceder a la primera, **Ctrl + Alt + F3** para la tercera y así hasta F6 para la sexta.

Para regresar al entorno gráfico: **Ctrl + Alt + F2**

- <https://tecnosoftware.com/es/terminal-tty-en-gnulinux/>

y cuáles son los archivos que las representan (tip: busque en el directorio **/dev**).//

```
maxi@maxi-VirtualBox:~$ cd /dev/
maxi@maxi-VirtualBox:/dev$ ls
autofs      cpu      fd      input   loop14  loop7    null    rtc    snapshot  tty10  tty19  tty27  tty35  tty43  tty51  tty6    tty80  tty817  tty825  tty85  userio  vcs6
clock      cpu_dma_latency full    kmsg    loop15  loop8    nvram   rtc0   snd     tty11  tty2   tty28  tty36  tty44  tty52  tty60  tty81  tty818  tty826  tty86  vboxguest  vcsa
bsg        cuse      fuse    log     loop16  loop9    port    sda    sr0    tty12  tty20  tty29  tty37  tty45  tty53  tty61  tty80  tty819  tty827  tty87  vboxuser   vcsa1
btrfs-control disk    hidraw0 loop0   loop17  loop-control  ppp    sda1  stderr  tty13  tty21  tty3   tty38  tty46  tty54  tty62  tty81  tty82  tty828  tty88  vcs   vcsa2
bus        dma_heap  hpet   loop1   loop2   mapper   psaux  sda2  stdin  tty14  tty22  tty30  tty39  tty47  tty55  tty63  tty812  tty820  tty829  tty89  vcs1  vcsa3
cdrom     dri      hugepages loop10  loop3   mcelog   ptmx   sda5  stdout  tty15  tty23  tty31  tty4   tty48  tty56  tty7   tty813  tty821  tty833  udmabuf  vcs2  vcsa4
char      dvd      hurng   loop11  loop4   mem     pts    sg0   tty   tty16  tty24  tty32  tty40  tty49  tty57  tty8   tty814  tty822  tty830  uhid  vcs3  vcsa5
console   encryptfs i2c-0   loop12  loop5   mqueue  random  sg1   tty0  tty17  tty25  tty33  tty41  tty55  tty58  tty9   tty815  tty823  tty831  uinput  vcs4  vcsa6
core      fb0      initctl loop13  loop6   net     rfkill  shm   tty1   tty18  tty26  tty34  tty42  tty50  tty59  ttyprintk  tty816  tty824  tty84  urandom  vcs5  vcsu
```

2.3.7. Use el comando **logout** para cerrar las sesiones abiertas en todas las terminales virtuales y vuelva a la **sesión gráfica** (<ctrl> <alt> F2 para Ubuntu).

```
Ubuntu 20.04.4 LTS maxi-VirtualBox tty3
maxi-VirtualBox login: maxi
Password:
Welcome to Ubuntu 20.04.4 LTS (GNU/Linux 5.15.0-43-generic x86_64)

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

Se pueden aplicar 82 actualizaciones de forma inmediata.
Para ver estas actualizaciones adicionales ejecute: apt list --upgradable

Your Hardware Enablement Stack (HWE) is supported until April 2025.
Last login: Fri Aug  5 23:24:23 -03 2022 on tty6
maxi@maxi-VirtualBox:~$ logout
```

```
Ubuntu 20.04.4 LTS maxi-VirtualBox tty3
maxi-VirtualBox login: _
```

De esta manera (**logout** o **exit**) te permite cerrar una terminal.

Si tienes 4 terminales abiertas y usas **logout**, solamente se cerrará la que utilizaste el comando, las otras 3 terminales permanecen prendidas hasta que ingreses una por una el comando **logout**.

```
=====
```

3. AYUDA

3.1. **INFO:** Info es un programa para leer documentación. Este se compone de una estructura del tipo árbol, dividido en nodos de información. Cada nodo describe un específico tema con un determinado nivel de detalle.

DATO= Cuando entras a **info**, apretas **H**, aparece un menú con todos los comandos que te permite ingresar dentro del **info**

```
'H' lists all Info commands;
```

```
-----Info: (dir)Top, 212 lines --Top-----
Teclas de las órdenes básicas para el manejo de Info

H      Salir de esta ventana de ayuda.
q      Salir completamente de Info.
h      Invocar el tutorial de uso de Info.

Up     Retroceder una línea.
Down   Avanzar una línea.
PgUp   Retroceder una página.
PgDn   Avanzar una página.
Home   Ir al principio de este nodo.
End    Ir al final de este nodo.

TAB    Ir al siguiente enlace de hiper-texto.
RET    Seguir el enlace de hiper-texto bajo el cursor.
l      Ir al último nodo visitado en esta ventana.

[      Ir al nodo anterior en este documento.
]      Ir al nodo siguiente en este documento.
p      Ir al nodo anterior en este nivel.
n      Ir al nodo siguiente en este nivel.
u      Ir al nodo un nivel más arriba.

-----Info: *Info Help*, 252 lines --Top-----
```

y para ver los demás comandos puedes bajar usando las flechas=



y para salir del menú vuelves a apretar H.

3.1.1. Ingrese a info y responda:

3.1.1.1. ¿Cómo se llama el **nodo raíz de Info**?

- Este es el menú principal de información, también conocido como (**directory node**)

3.1.1.2. Ubique el cursor en la línea (* cp:) y presione **ENTER**.

3.1.1.3. ¿Qué sucedió?

- una vez que encontré el (*cp) y se apreta **ENTER**, muestra una lista de toda la información que lo que hace **cp**

3.1.1.4. ¿Cómo se llama este **nodo**?

- Con el comando **CP** (abreviatura de “copy”) puedes hacer **copias** de archivos individuales o de carpetas enteras.

3.1.1.5. ¿Cuál es el próximo nodo?

empezando desde **CP** vas bajando hasta encontrar próximo nodo que es **dd**

```
11.2 'dd': Convert and copy a file
=====
```

- **dd** : Convert and copy a file

3.1.1.6. ¿Cómo puedo moverme al próximo nodo?
de la siguiente manera:

con teclas de direcciones



- se usa la tecla de dirección pulsando la tecla ↓

3.1.1.7. ¿Cómo puedo moverme al nodo anterior?

- con la tecla de dirección pulsando la tecla ↑

y también presionando la tecla L

3.1.2. Presione la tecla 'u'.

3.1.2.1. ¿Qué sucedió?

- si estás dentro de algún **nodo** y apretas U te envía a otra página que es la “descripción de comandos para la manipulación básica de archivos”.

```
Next: Special file types,  Prev: Directory listing,  Up: Top
11 Basic operations
*****
This chapter describes the commands for basic file manipulation:
copying, moving (renaming), and deleting (removing).

* Menu:
  * cp invocation::          Copy files.
  * dd invocation::          Convert and copy a file.
  * install invocation::    Copy files and set attributes.
  * mv invocation::          Move (rename) files.
  * rm invocation::          Remove files or directories.
  * shred invocation::      Remove files more securely.
```

pero si no estás en ningún **nodo** te informa este mensaje:

```
Este nodo no contiene enlace alguno a 'Up'
```

3.1.2.2. ¿En qué nodo se encuentra? REVISAR

nos encontramos en el menú de “**basic operations**” al apretar u

```
Next: Special file types,  Prev: Directory listing,  Up: Top  
11 Basic operations  
*****  
This chapter describes the commands for basic file manipulation:  
copying, moving (renaming), and deleting (removing).  
* Menu:  
* cp invocation::          Copy files.  
* dd invocation::          Convert and copy a file.  
* install invocation::    Copy files and set attributes.  
* mv invocation::          Move (rename) files.  
* rm invocation::          Remove files or directories.  
* shred invocation::      Remove files more securely.
```

Pero antes nos encontrábamos en el nodo:

```
11.2 'dd': Convert and copy a file  
• =====
```

3.1.3. Repita el punto 3.1.2. hasta que llegue a la raíz de Info.

3.1.3.1. ¿Con qué tecla puedo volver directamente a este nodo?.

- con la tecla **L**, te permite retroceder atrás

3.1.3.2. ¿Cuál es el método directo para acceder al nodo **cp**? (tip: sin desplazar el cursor).

- con la **lupa** de buscar, que se encuentra en la esquina superior derecha, y se accede a ella dandole click con el ratón.

y también presionando la tecla **control+R** y escribes lo que quieras buscar y apretas **ENTER**.

3.1.4. ¿Cómo puedo buscar una palabra clave dentro de un nodo?

- escribiendo la palabra dentro de la **lupa**.

```
/ (search)          Leer una cadena de texto y buscarla
```

o también apretas la tecla **control+S** y escribes lo que quieras buscar, si vuelves a apretar **control+S** la palabra buscada pero la siguiente.

o también apretas la tecla **control+R** y escribes lo que quieras buscar, si vuelves a apretar **control+R** la palabra buscada pero la anteriormente.

3.1.5. ¿Cómo puedo buscar la siguiente palabra clave, buscada anteriormente?

```
Buscar anterior      Mayús+Ctrl+H
```

apretas la tecla **control+R** y escribes lo que quieras buscar, si vuelves a apretar **control+R** la palabra buscada pero la anteriormente.

3.1.6. ¿Cómo puedo salir de **Info**? -hágalo. clave

- pulsando la tecla **Q** o también presionando la tecla **control+c**.

3.2. **MAN**: **man** es un programa que formatea y muestra las páginas del **manual**.

3.2.1. ¿Cuál es la diferencia entre **man** e **info**?

- **man** es la forma tradicional de ayuda para casi todos los comandos de su sistema, y no solo, a veces también los archivos de configuración tienen su propia página de manual.

- **info** es un sistema alternativo para proporcionar páginas de manual para comandos, basado en GNU emacs. Se proporciona principalmente para comandos y utilidades GNU.

https://www.compuhoy.com/cual-es-el-uso-del-comando-man-e-info-en-linux/#_Cual_es_la_diferencia_entre_man_e_info

3.2.2. ¿Cómo puedo ver la **información** de un determinado comando?

- **info** [comando]

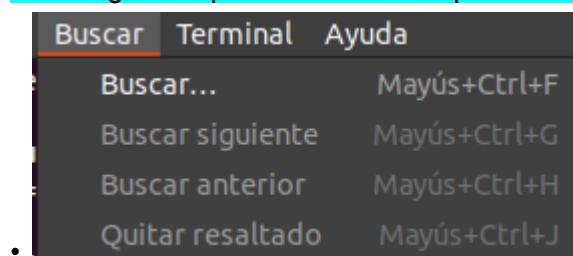
ejemplo: **maxi@maxi-VirtualBox:~\$ info man**

Dato:

```
maxi@maxi-VirtualBox:~$ info cd
info: No existe ningún elemento de menú llamado 'cd' en el nodo '(dir)Top'
```

tiene que estar dentro de **info** si no, no te podrá mostrar **información**.

3.2.3. ¿Cómo puedo buscar una palabra clave dentro de la página del **manual**?



para poder buscar dentro del manual también se puede ingresar el comando “**ESC I**” y escribir lo que quieras buscar y te lo encontrara

3.2.4. ¿Cómo puedo salir?

- pulsando la tecla **Q**

3.2.5. ¿Cómo hago para buscar una palabra clave determinada en todas las páginas del manual?

/ (search) Leer una cadena de texto y buscarla

- para poder buscar una palabra clave de forma general y para cada nodo se tiene que apretar las teclas “**ESC /**” y escribes lo que quieres buscar

3.2.6. ¿Qué es lo sucede al realizar lo siguiente?

3.2.6.1. man

```
maxi@maxi-VirtualBox:~$ man
¿Qué página de manual quiere?
• Por ejemplo, pruebe 'man man'.
```

3.2.6.2. man man

- abre la **interfaz** de los manuales de referencia del sistema de **man** y sus **SINOPSIS**

```
MAN(1)                               Utilidades del paginador del manual                               MAN(1)

NOMBRE
    man - interfaz de los manuales de referencia del sistema

SYNOPSIS
    man [opciones de man] [[sección] página ...]
    man -k [opciones de apropos] regexp ...
    man -s [opciones de man] [sección] term ...
    man -f [opciones de man] página ...
    man -l [opciones de man] fichero ...
    man -w|-W [opciones de man] página ...

DESCRIPCIÓN
    man es el paginador de manuales del sistema. Cada argumento de página argumento dado a man normalmente es el nombre de un programa, utilidad o función. La página del manual asociada con cada uno de estos argumentos es, pues, encontrada y mostrada. Si se proporciona una sección, man mirará solo en esa sección del manual. La acción predeterminada es buscar en todas las secciones disponibles siguiendo un orden predefinido (véase DEFAULTS), y mostrar solo la primera página encontrada, incluso si la página existe en varias secciones.

    La tabla de abajo muestra los números de sección del manual seguidos por los tipos de página que contienen.

    1 Programas ejecutables u órdenes de la shell
    2 Órdenes al sistema (nombres proporcionados por el núcleo)
    3 Llamadas a biblioteca (funciones dentro de bibliotecas de programa)
    4 Ficheros especiales (normalmente se encuentran en /dev)
    5 Fornatos de fichero y convenios, p.e. /etc/passwd
    6 Juegos
    7 Miscelánea (incluidos paquetes de macros y convenios), p.e. man(7), groff(7)
    8 Órdenes de administración del sistema (normalmente solo para root)
    9 Rutinas del núcleo [No estándar]

    Una página de manual contiene varias secciones.

    Nombres de sección convencionales: NOMBRE, SINOPSIS, CONFIGURACIÓN, DESCRIPCIÓN, OPCIONES, ESTADO DE SALIDA, VALOR DEVUELTO, ERRORES, ENTORNO, FICHEROS, VERSIONES, CONFORME A, NOTAS, DEFECTOS, EJEMPLO, AUTORES, y VÉASE TAMBÉN.

    Los siguientes convenios se aplican a la sección SINOPSIS y pueden utilizarse como guía en otras secciones.

    escritura resaltada   teclea exactamente como se muestra.
    texto en cursiva   sustituye con argumento apropiado.
    [-abc]              cualquiera de los argumentos dentro de [ ] o todos ellos son opcionales.
    -a|-b              las opciones delimitadas por | no pueden usarse a la vez.
    argumento ...       argumento es repetible.

Manual page man(1) line 3 (press h for help or q to quit)
```

3.2.6.3. man cp

```
CP(1)                               User Commands                               CP(1)

NAME
    cp - copy files and directories

SYNOPSIS
    cp [OPTION]... [-T] SOURCE DEST
    cp [OPTION]... SOURCE... DIRECTORY
    cp [OPTION]... -t DIRECTORY SOURCE...

DESCRIPTION
    Copy SOURCE to DEST, or multiple SOURCE(s) to DIRECTORY.

    Mandatory arguments to long options are mandatory for short options too.

    -a, --archive
        same as -dR --preserve=all

    --attributes-only
        don't copy the file data, just the attributes

    Ayuda
    --backup[=CONTROL]
        make a backup of each existing destination file

    -b
        like --backup but does not accept an argument

    --copy-contents
        copy contents of special files when recursive

    -d
        same as --no-dereference --preserve=links

    -f, --force
        if an existing destination file cannot be opened, remove it and try again (this option is ignored when the -n option is also used)

    -i, --interactive
        prompt before overwrite (overrides a previous -n option)

    -H
        follow command-line symbolic links in SOURCE

    -l, --lnk
        hard link files instead of copying

Manual page cp(1) line 1 (press h for help or q to quit)
```

3.2.6.4. man printf

```
PRINTF(1)                               User Commands                         PRINRF(1)

NAME
    printf - format and print data

SYNOPSIS
    printf FORMAT [ARGUMENT]...
    printf OPTION

DESCRIPTION
    Print ARGUMENT(s) according to FORMAT, or execute according to OPTION:
    --help display this help and exit
    --version          output version information and exit
    FORMAT controls the output as in C printf. Interpreted sequences are:
    \"
    \\
    \a    alert (BEL)
    \b    backspace
    \c    produce no further output
    \e    escape
    \f    form feed
    \n    new line
    \r    carriage return
    \t    horizontal tab
    \v    vertical tab
    \NNN   byte with octal value NNN (1 to 3 digits)
Manual page printf(1) line 1 (press h for help or q to quit)
```

3.2.6.5. man printf

```
PRINRF(3)                         Linux Programmer's Manual                     PRINRF(3)

NAME
    printf, fprintf, dprintf, sprintf, snprintf, vprintf, vfprintf, vd-
    printf, vsprintf, vsnprintf - formatted output conversion

SYNOPSIS
    #include <stdio.h>

    int printf(const char *format, ...);
    int fprintf(FILE *stream, const char *format, ...);
    int dprintf(int fd, const char *format, ...);
    int sprintf(char *str, const char *format, ...);
    int snprintf(char *str, size_t size, const char *format, ...);

    #include <stdarg.h>

    int vprintf(const char *format, va_list ap);
    int vfprintf(FILE *stream, const char *format, va_list ap);
    int vdprintf(int fd, const char *format, va_list ap);
    int vsprintf(char *str, const char *format, va_list ap);
    int vsnprintf(char *str, size_t size, const char *format, va_list ap);

Feature Test Macro Requirements for glibc (see feature_test_macros(7)):
    snprintf(), vsnprintf():
        _XOPEN_SOURCE >= 500 || _ISOC99_SOURCE ||
        /* Glibc versions <= 2.19: */ _BSD_SOURCE

    dprintf(), vdprintf():
        Since glibc 2.10:
            _POSIX_C_SOURCE >= 200809L
        Before glibc 2.10:
            _GNU_SOURCE

DESCRIPTION
    The functions in the printf() family produce output according to a for-
    mat as described below. The functions printf() and vprintf() write
    output to stdout, the standard output stream; fprintf() and vfprintf()
    write output to the given output stream; sprintf(), snprintf(),
    vsprintf() and vsnprintf() write to the character string str.
    The function dprintf() is the same as fprintf() except that it outputs
• Manual page fprintf(3) line 1 (press h for help or q to quit)
```

3.2.6.6. man sprintf

```
PRINTF(3)                               Linux Programmer's Manual                               PRINTF(3)

NAME
    printf, fprintf, dprintf, sprintf, snprintf, vprintf, vfprintf, vdprintf, vsprintf, vsnprintf - formatted output conversion

SYNOPSIS
#include <stdio.h>

int printf(const char *format, ...);
int fprintf(FILE *stream, const char *format, ...);
int dprintf(int fd, const char *format, ...);
int snprintf(char *str, const char *format, ...);
int vsnprintf(char *str, size_t size, const char *format, ...);

#include <stdarg.h>

int vprintf(const char *format, va_list ap);
int vfprintf(FILE *stream, const char *format, va_list ap);
int vdprintf(int fd, const char *format, va_list ap);
int vsprintf(char *str, const char *format, va_list ap);
int vsnprintf(char *str, size_t size, const char *format, va_list ap);

Feature Test Macro Requirements for glibc (see feature_test_macros(7)):

snprintf(), vsnprintf():
    _XOPEN_SOURCE >= 500 || __ISOC99_SOURCE ||
    || /* Glibc versions <= 2.19: */ __BSD_SOURCE

dprintf(), vdprintf():
    Since glibc 2.10:
        _POSIX_C_SOURCE >= 200809L
    Before glibc 2.10:
        _GNU_SOURCE

DESCRIPTION
The functions in the printf() family produce output according to a format as described below. The functions printf() and vprintf() write output to stdout, the standard output stream; fprintf() and vfprintf() write output to the given output stream; snprintf(), vsprintf() and vsnprintf() write to the character string str. The function dprintf() is the same as fprintf() except that it outputs to a file descriptor, fd, instead of to a stdio stream. The functions sprintf() and vsnprintf() write at most size bytes (including the terminating null byte ('\0')) to str. The functions vprintf(), vfprintf(), vdprintf(), vsprintf(), vsnprintf() are equivalent to the functions printf(), fprintf(), dprintf(), sprintf(), respectively, except that they
Manual page sprintf(3) line 1 (press h for help or q to quit)
```

3.2.6.7. man 3 printf

```
PRINTF(3)                               Linux Programmer's Manual                               PRINTF(3)

NAME
    printf, fprintf, dprintf, sprintf, snprintf, vprintf, vfprintf, vdprintf, vsprintf, vsnprintf - formatted output conversion

SYNOPSIS
#include <stdio.h>

int printf(const char *format, ...);
int fprintf(FILE *stream, const char *format, ...);
int dprintf(int fd, const char *format, ...);
int snprintf(char *str, const char *format, ...);
int vsnprintf(char *str, size_t size, const char *format, ...);

#include <stdarg.h>

int vprintf(const char *format, va_list ap);
int vfprintf(FILE *stream, const char *format, va_list ap);
int vdprintf(int fd, const char *format, va_list ap);
int vsprintf(char *str, const char *format, va_list ap);
int vsnprintf(char *str, size_t size, const char *format, va_list ap);

Aurelio Test Macro Requirements for glibc (see feature_test_macros(7)):

snprintf(), vsnprintf():
    _XOPEN_SOURCE >= 500 || __ISOC99_SOURCE ||
    || /* Glibc versions <= 2.19: */ __BSD_SOURCE

dprintf(), vdprintf():
    Since glibc 2.10:
        _POSIX_C_SOURCE >= 200809L
    Before glibc 2.10:
        _GNU_SOURCE

DESCRIPTION
The functions in the printf() family produce output according to a format as described below. The functions printf() and vprintf() write output to stdout, the standard output stream; fprintf() and vfprintf() write output to the given output stream; snprintf(), vsprintf() and vsnprintf() write to the character string str. The function dprintf() is the same as fprintf() except that it outputs to a file descriptor, fd, instead of to a stdio stream. The functions sprintf() and vsnprintf() write at most size bytes (including the terminating null byte ('\0')) to str. The functions vprintf(), vfprintf(), vdprintf(), vsprintf(), vsnprintf() are equivalent to the functions printf(), fprintf(), dprintf(), sprintf(), respectively, except that they
Manual page printf(3) line 1 (press h for help or q to quit)
```

3.2.7. Si los comandos del punto anterior le dieron un mensaje de error, instale el paquete **manpages-dev** y vuelva a ejecutarlos (utilice **sudo apt install manpages-dev** si está usando Ubuntu).

- escribir en la consola “**sudo apt install manpages-dev**”

3.2.8. Del punto anterior, responder:

3.2.8.1. Al invocar **man** junto con **fprintf** y **sprintf** muestra la misma página. ¿Porqué no muestra la misma página al invocarlo con **printf**?. (TIP: vea el punto 3.2.6.2).

- No muestra lo mismo por que al invocar “**man** **sprintf** y **man** **fprintf**”, corresponden a la sección 3 de **printf**, y al invocar “**man** **printf**”, corresponden a la sección 1 de **printf**, no existe sección 2 de **printf**

3.2.8.2. ¿Cómo puedo invocar al **man** para ver directamente la función **printf** del lenguaje C?.

```
maxi@maxi-VirtualBox:~$ man printf
```

```
PRINTF(1)                               User Commands                               PRINTF(1)

NAME
    printf - format and print data

SYNOPSIS
    printf FORMAT [ARGUMENT]...
    printf OPTION

DESCRIPTION
    Print ARGUMENT(s) according to FORMAT, or execute according to OPTION:
    --help display this help and exit
    LibreOfficeWriter
        output version information and exit

    FORMAT controls the output as in C printf. Interpreted sequences are:
    \"
    \\
    \a    alert (BEL)
    \b    backspace
    \c    produce no further output
    \e    escape
    \f    form feed
    \n    new line
    \r    carriage return
    \t    horizontal tab
    \v    vertical tab
    \NNN   byte with octal value NNN (1 to 3 digits)

Manual page printf(1) line 1 (press h for help or q to quit)
```

3.3. HELP: **help** es la ayuda que ofrece el shell de GNU/LINUX para utilizar sus comandos.

3.3.1. ¿Cuál es la diferencia entre **help** e **info**?

- **help** es un comando integrado en el bash shell (y solo en ese shell) que documenta algunos de los comandos integrados y palabras clave de ese shell. Ese es un sistema de documentación interna de ese caparazón. Otros shells tienen su propio sistema de documentación (ksh93tiene --helpy --manopciones para sus componentes integrados, zshtiene un asistente de ayuda de ejecución que extrae información de manuales en otros formatos). Otros comandos vimtienen su propio sistema de documentación incorporado.

info es otro sistema de documentación originario del proyecto GNU. Es hipertexto con enlaces (anterior a la web). Un manual de información es como un libro digital con un concepto de tabla de contenido e índice (de búsqueda) que ayuda a localizar la información.

<https://unix.stackexchange.com/questions/19451/difference-between-help-info-and-man-command>

3.3.2. ¿Cuál es la diferencia entre **help** y **man**?

help = El comando **help** guarda y muestra información sobre los comandos de bash. Si desconocemos y tenemos dudas sobre cómo usar un comando, tipeando en la terminal **-help** nos va a mostrar una lista incluyendo las opciones de un determinado comando.

man = La función del comando **man** es mostrar los manuales detallados para cada comando. Estos se conocen como «man-pages». Las páginas de **man** generalmente contienen una información mucho más detallada que obtendrás con la opción **-h** o **-help**.

<https://www.profesionalreview.com/2016/07/23/ayuda-comandos-linux/#:~:text=help,opciones%20de%20un%20determinado%20comando.>

3.3.3. ¿Qué sucede al invocar al **help**?

Use «man -k» o «info» para saber más sobre las órdenes que no están en esta lista.

Un asterisco (*) junto a un nombre significa que la orden está desactivada.

```
id_trabajo [&]
(( expresión ))
. nombreArchivo [argumentos]
:
[ arg... ]
[[ expresión ]]
alias [-p] [nombre[=valor] ... ]
bg [id_trabajo ...]
bind [-lpsvPSVX] [-m comb_teclas] [-f]
break [n]
builtin [orden-interna-del-intérprete]
caller [expresión]
case PALABRA in [PATRÓN [| PATRÓN]...]
cd [-L][-P [-e]]] [dir]
command [-pVv] orden [arg ...]
compgen [-abcdefgjksuv] [-o opción] >
complete [-abcdefgjksuv] [-pr] [-DEI]>
compopt [-o]+o opción [-DEI] [nombre]>
continue [n]
coproc [NOMBRE] orden [redirecciones]>
declare [-aAfFgilrtux] [-p] [nombre[=>
dirs [-clpv] [+N] [-N]
disown [-h] [-ar] [idtrabajo ... | pi>
echo [-neE] [arg ...]
enable [-a] [-dnps] [-f archivo] [nom>
eval [arg ...]
exec [-cl] [-a nombre] [orden [argume>
exit [n]
export [-fn] [nombre[=valor] ... ] ó e>
false
fc [-e nombre_e] [-lnr] [primero] [último]
fg [id_trabajo]
for NOMBRE [in PALABRAS ... ] ; do ÓRDENES
for (( exp1; exp2; exp3 )); do ÓRDENES
function nombre { ÓRDENES ; } o nombr>
getopts cadena_opciones nombre [arg]
hash [-lr] [-p ruta] [-dt] [nombre ...]
help [-dms] [patrón ...]
```

```
history [-c] [-d despl] [n] ó histor>
if ÓRDENES; then ÓRDENES; [ elif ÓRDENES
jobs [-lnprs] [idtrabajo ...] o jobs>
kill [-s id_señal | -n num_señal | ->
let arg [arg ...]
local [opción] nombre[=valor] ...
logout [n]
mapfile [-d delim] [-n cuenta] [-0 o>
popd [-n] [+N | -N]
printf [-v var] formato [argumentos]
pushd [-n] [+N | -N | dir
pwd [-LP]
read [-ers] [-a matriz] [-d delim] [>
readarray [-d delim] [-n cuenta] [-0 o>
readonly [-aAf] [nombre[=valor] ...]>
return [n]
select NOMBRE [in PALABRAS ... ;] do>
set [-abefhkmnpptuvwxyzBCHP] [-o nombre->
shift [n]
shopt [-pqsu] [-o] [nombre_opción...]
source nombreArchivo [argumentos]
suspend [-f]
test [expresión]
time [-p] tubería
times
trap [-lp] [[arg] id_señal ...]
true
type [-afptP] nombre [nombre ...]
typeset [-aAfFgilnrtux] [-p] nombre[>
ulimit [-SHabcdefiklmnpqrstuvwxyzPT] [l]>
umask [-p] [-S] [modo]
unalias [-a] nombre [nombre ...]
unset [-f] [-v] [-n] [nombre ...]
until ÓRDENES; do ÓRDENES; done
variables - Nombres y significados de variables
wait [-fn] [id ...]
while ÓRDENES; do ÓRDENES; done
{ ÓRDENES ; }
```

3.3.4. ¿Cómo puedo ver la información de un determinado comando?

- tienes que poner el comando y luego **-help**, ejemplo =

```
maxi@maxi-VirtualBox:~$ dd --help
```

```
Modo de empleo: dd [OPERANDO]...
    o bien: dd OPCIÓN
Copy a file, converting and formatting according to the operands.

bs=BYTES      read and write up to BYTES bytes at a time (default: 512);
               overrides ibs and obs
cbs=BYTES     convert BYTES bytes at a time
conv=CONVS    convert the file as per the comma separated symbol list
count=N       copy only N input blocks
ibs=BYTES     read up to BYTES bytes at a time (default: 512)
if=FILE       read from FILE instead of stdin
iflag=FLAGS   read as per the comma separated symbol list
obs=BYTES     write BYTES bytes at a time (default: 512)
of=FILE       write to FILE instead of stdout
oflag=FLAGS   write as per the comma separated symbol list
seek=N        skip N obs-sized blocks at start of output
skip=N        skip N ibs-sized blocks at start of input
status=LEVEL  The LEVEL of information to print to stderr;
               'none' suppresses everything but error messages,
               'noxfer' suppresses the final transfer statistics,
               'progress' shows periodic transfer statistics
```

N and BYTES may be followed by the following multiplicative suffixes:
C =1, W =2, B =512, kB =1000, K =1024, MB =1000*1000, M =1024*1024, xM =M,
GB =1000*1000*1000, G =1024*1024*1024, and so on for T, P, E, Z, Y.

Each CONV symbol may be:

ascii	de EBCDIC a ASCII
ebcdic	de ASCII a EBCDIC
ibm	de ASCII a EBCDIC alternado
block	rellena los registros terminados en nueva linea con espacios hasta el tamaño de cbs
unblock	sustituye los espacios que sobran en los registros de tamaño cbs con un carácter de nueva linea
lcase	cambia las mayúsculas a minúsculas
ucase	cambia las minúsculas a mayúsculas
sparse	intenta efectuar seek en lugar de escribir el resultado para los bloques de entrada nulos
swab	intercambia cada pareja de bytes de entrada
sync	rellena cada bloque de entrada con NULs hasta el tamaño de ibs; cuando se utiliza con block o unblock, rellena con espacios en lugar de con NULos

3.3.5. ¿Qué sucede al utilizar el comando **help cd**?

•

```
maxi@maxi-VirtualBox:~$ help cd
cd: cd [-L|[-P [-e]]] [dir]
      Modifica el directorio de trabajo del shell.

      Modifica el directorio actual a DIR. DIR por defecto es el valor de la
      variable de shell HOME.

      La variable CDPATH define la ruta de búsqueda para el directorio que
      contiene DIR. Los nombres alternativos de directorio en CDPATH se
      separan con dos puntos (:). Un nombre de directorio nulo es igual que
      el directorio actual. Si DIR comienza con una barra inclinada (/),
      entonces no se usa CDPATH.

      Si no se encuentra el directorio, y la opción del shell 'cdable_vars'
      está activa, entonces se trata la palabra como un nombre de variable.
      Si esa variable tiene un valor, se utiliza su valor para DIR.

      Opciones:
      -L      fuerza a seguir los enlaces simbólicos: resuelve los enlaces
              simbólicos en DIR después de procesar las instancias de `..'
      -P      usa la estructura física de directorios sin seguir los enlaces
              simbólicos: resuelve los enlaces simbólicos en DIR antes de procesar
              las instancias de `..'
      -e      si se da la opción -P y el directorio actual de trabajo no se
              puede determinar con éxito, termina con un estado diferente de cero.

      La acción por defecto es seguir los enlaces simbólicos, como si se
      especificara '-L'.
      `..' se procesa quitando la componente del nombre de la ruta inmediatamente
      anterior hasta una barra inclinada o el comienzo de DIR.

      Estado de Salida:
      Devuelve 0 si se cambia el directorio, y si $PWD está definido como
      correcto cuando se emplee -P; de otra forma es diferente a cero.
```

3.4. whereis

3.4.1. ¿Qué sucede al utilizar el comando **whereis cp**?

```
maxi@maxi-VirtualBox:~$ whereis cp
cp: /usr/bin/cp /usr/share/man/man1/cp.1.gz
maxi@maxi-VirtualBox:~$
```

3.4.2. ¿Qué es la información que se muestra por pantalla al ejecutar el punto anterior?

- muestra la **localización** del archivo, que en este caso sería **CP**.

3.4.3. ¿Qué ocurre si se ejecuta **whereis *** sobre un directorio? (Tip: si no pasa nada, intentelo nuevamente pero primero ejecute **cd /bin**)

```
maxi@maxi-VirtualBox:~$ whereis *
```

asdasd:
Descargas:
Documentos::
Escritorio:
Imágenes:
initial shift state, if any:
miprimerprog:
miprogs:
Música:
Plantillas:
prueba:
prueba:
Público:
snap: /usr/bin/snap /usr/share/man/man8/snap.8.gz
tream:
Videos:
maxi@maxi-VirtualBox:~\$

```
maxi@maxi-VirtualBox:~$ cd /bin  
maxi@maxi-VirtualBox:/bin$ whereis *  
[...]  
aa-enabled: /usr/bin/aa-enabled /usr/share/man/man1/aa-enabled.1.gz  
aa-exec: /usr/bin/aa-exec /usr/share/man/man1/aa-exec.1.gz  
aconnect: /usr/bin/aconnect /usr/share/man/man1/aconnect.1.gz  
acp1_listen: /usr/bin/acp1_listen /usr/share/man/man8/acp1_listen.8.gz  
addapt-repository: /usr/bin/add-apt-repository /usr/share/man/man1/add-apt-repository.1.gz  
addpart: /usr/bin/addpart /usr/share/man/man1/addpart.8.gz  
alsa-base: /usr/bin/alsa-base /usr/share/man/man1/alsa-base.1.gz  
alsabat: /usr/bin/alsabat /usr/share/man/man1/alsabat.1.gz  
alsa-loop: /usr/bin/alsa-loop /usr/share/man/man1/alsa-loop.1.gz  
alsamixer: /usr/bin/alsamixer /usr/share/man/man1/alsamixer.1.gz  
alsaplug: /usr/bin/alsaplug /usr/share/man/man1/alsaplug.1.gz  
alsaucm: /usr/bin/alsaucm /usr/share/man/man1/alsaucm.1.gz  
amidi: /usr/bin/amidi /usr/share/man/man1/amidi.1.gz  
amixer: /usr/bin/amixer /usr/share/man/man1/amixer.1.gz  
amufORMAT: /usr/bin/amufORMAT.sh  
apg: /usr/bin/apg /usr/lib/apg /etc/apg.conf /usr/share/man/man1/apg.1.gz  
apgbfm: /usr/bin/apgbfm /usr/share/man/man1/apgbfm.1.gz  
aplay: /usr/bin/aplay /usr/share/man/man1/aplay.1.gz  
aplaymidi: /usr/bin/aplaymidi /usr/share/man/man1/aplaymidi.1.gz  
apport: /usr/bin/apport /usr/share/man/man1/apport.1.gz  
apport-build: /usr/bin/apport-build /usr/share/man/man1/apport-build.1.gz  
apport-ccli: /usr/bin/apport-ccli /usr/share/man/man1/apport-ccli.1.gz  
apport-collect: /usr/bin/apport-collect /usr/share/man/man1/apport-collect.1.gz  
apport-unpack: /usr/bin/apport-unpack /usr/share/man/man1/apport-unpack.1.gz  
appres: /usr/bin/appres /usr/share/man/man1/appres.1.gz  
appstreamcli: /usr/bin/appstreamcli /usr/share/man/man1/appstreamcli.1.gz  
apt: /usr/bin/apt /lib/apt /etc/apt /usr/share/man/man1/apt.8.gz  
apt-add-repository: /usr/bin/apt-add-repository /usr/share/man/man1/apt-add-repository.1.gz  
apt-cache: /usr/bin/apt-cache /usr/share/man/man8/apt-cache.8.gz  
apt-cdrom: /usr/bin/apt-cdrom /usr/share/man/man8/apt-cdrom.8.gz  
apt-config: /usr/bin/apt-config /usr/share/man/man8/apt-config.8.gz  
apt-extracttemplates: /usr/bin/apt-extracttemplates /usr/share/man/man1/apt-extracttemplates.1.gz  
apt-farchive: /usr/bin/apt-farchive /usr/share/man/man1/apt-farchive.1.gz  
apt-get: /usr/bin/apt-get /usr/share/man/man8/apt-get.8.gz  
apt-key: /usr/bin/apt-key /usr/share/man/man8/apt-key.8.gz  
apt-mark: /usr/bin/apt-mark /usr/share/man/man8/apt-mark.8.gz  
apt-sortpkgs: /usr/bin/apt-sortpkgs /usr/share/man/man1/apt-sortpkgs.1.gz  
aptutil-gtk: /usr/bin/aptutil-gtk /usr/share/man/man8/aptutil-gtk.8.gz  
ar: /usr/bin/ar /usr/share/ar.h /usr/share/man/man1/ar.1.gz  
arch: /usr/bin/arch /usr/share/man/man1/arch.1.gz /usr/src/linux-hwe-5.15-headers-5.15.0-46/arch /usr/src/linux-hwe-5.15-headers-5.15.0-43/arch /usr/src/linux-hwe-5.13-headers-5.13.0-30/arch /usr/src/linux-headers-5.15.0-43-generic/arch  
arecord: /usr/bin/arecord /usr/share/man/man1/arecord.1.gz  
arecordmidi: /usr/bin/arecordmidi /usr/share/man/man1/arecordmidi.1.gz
```

3.4.4. ¿Cuál es la diferencia entre **whereis** y **find**?

- El programa **find** busca en un árbol de directorios para encontrar un archivo o grupo de archivos. Atraviesa el árbol de directorios e informa todas las ocurrencias de un archivo que coincida con las especificaciones del usuario. El programa de búsqueda incluye una capacidad de búsqueda muy potente.
- el **whereis** es un comando que simplemente devuelve la ubicación de los ejecutables, las páginas man y las fuentes de un programa (ver man whereis)

<https://askubuntu.com/questions/832562/difference-among-whereis-locate-and-findcommand>

3.5. whatis

- **whatis:** lo que hace es explicar lo que hace un nodo
ejemplo:

```
maxi@maxi-VirtualBox:~$ whatis man
man (1)           - interfaz de los manuales de referencia del sistema
man (7)           - macros to format man pages
```

3.5.1. ¿Qué sucede al utilizar el comando **whatis cp?**

```
maxi@maxi-VirtualBox:~$ whatis cp
cp (1)           - copy files and directories
maxi@maxi-VirtualBox:~$
```

3.5.2. Cambie al directorio **/bin**, y ejecute el comando **whatis *** ¿Qué ocurrió?

```
maxi@maxi-VirtualBox:~$ cd /bin
maxi@maxi-VirtualBox:/bin$
```

```
maxi@maxi-VirtualBox:/bin$ whatis *
apropos (1)           - busca nombres y descripciones de páginas de manual
apt-cache (8)          - Realiza consultas al caché de APT
apt-cdrom (8)          - Herramienta de APT para la gestión de discos ópticos
apt-config (8)          - Programa para consultar la configuración de APT
apt-extracttemplates (1) - Herramienta de extracción de configuración de ...
apt-ftparchive (1)      - Herramienta para generar ficheros de índice
apt-sortpkgs (1)        - Herramienta para ordenar los ficheros de índice de pa...
catman (8)              - crea o actualiza las páginas de manual preformateadas
debconf (1)              - Ejecuta un programa que hace uso de debconf
debconf-apt-progress (1) - Instala paquetes usando debconf para mostrar una b...
debconf-communicate (1) - Permite la comunicación con debconf
debconf-copydb (1)       - Copia una base de datos de debconf
debconf-escape (1)        - Asistente para la interacción con la funcionalidad de...
debconf-set-selections (1) - insert new values into the debconf database
debconf-show (1)          - Realiza consultas a la base de datos de debconf
dpkg-split (1)            - Herramienta para separar y unir paquetes Debian
lexgrog (1)               - analiza la información de cabecera en páginas man
man (1)                  - interfaz de los manuales de referencia del sistema
mandb (8)                 - crea o actualiza las cachés de índices de páginas ...
manpath (1)                - determina la ruta de búsqueda para páginas de manual
manpath (5)                - formato del fichero /etc/manpath.config
man-recode (1)              - convierte páginas de manual a otra codificación
run-parts (8)                - Ejecuta scripts o programas en un directorio
savelog (8)                 - Guarda un archivo de registro de eventos
sensible-editor (1)         - Facilita editar y leer ficheros así como explorar la ...
tempfile (1)                 - Crea un fichero temporal de forma segura
update-alternatives (1)       - Gestión de enlaces simbólicos que determinan órd...
whatis (1)                  - muestra descripciones de una línea de las páginas de...
which (1)                   - Busca una orden
[ (1)                      - check file types and compare values
aa-enabled (1)                - test whether AppArmor is enabled
aa-exec (1)                   - confine a program with the specified AppArmor profile
aconnect (1)                  - ALSA sequencer connection manager
acpi_listen (8)                - ACPI event listener
add-apt-repository (1)         - Adds a repository into the /etc/apt/sources.list or ...
addpart (8)                   - tell the kernel about the existence of a partition
alsabat (1)                   - command-line sound tester for ALSA sound card driver
alsaloop (1)                  - command-line PCM loopback
alsamixer (1)                  - soundcard mixer for ALSA soundcard driver, with ncurses...
alsatplg (1)                   - ALSA Topology Compiler
alsaucm (1)                   - ALSA Use Case Manager
amidi (1)                     - read from and write to ALSA RawMIDI ports
```

3.5.3. Utilice el comando **apropos passwd** y **whatis passwd**. Enumere las diferencias encontradas en el resultado de cada uno de los comandos.

```
maxi@maxi-VirtualBox:~$ apropos passwd
update-passwd (8)      - Actualiza /etc/passwd, /etc/shadow y /etc/group de forma segura
chgpasswd (8)           - update group passwords in batch mode
chpasswd (8)            - update passwords in batch mode
fgetpwent_r (3)         - get passwd file entry reentrantly
getpwent_r (3)          - get passwd file entry reentrantly
gpasswd (1)             - administer /etc/group and /etc/gshadow
grub-mkpasswd-pbkdf2 (1) - generate hashed password for GRUB
openssl-passwd (1ssl)   - compute password hashes
pam_localuser (8)       - require users to be listed in /etc/passwd
passwd (1)              - change user password
passwd (1ssl)           - compute password hashes
passwd (5)              - the password file
passwd2des (3)          - RFS password encryption
maxi@maxi-VirtualBox:~$ █
```

```
maxi@maxi-VirtualBox:~$ whatis passwd
passwd (1ssl)           - compute password hashes
passwd (1)              - change user password
passwd (5)              - the password file
maxi@maxi-VirtualBox:~$ █
```

la diferencias entre **apropos passwd** y **whatis passwd** es:

1. **apropos passwd** te da mayor información que **whatis passwd**
2. la información que ofrece **whatis passwd** está en **apropos passwd**

=====

4. DIRECTORIOS

4.1. ¿Para qué se usa el comando **cd**? Ejecute las siguientes variantes de **cd** e indique cuál fue el resultado obtenido:

4.1.1. **cd /**

```
maxi@maxi-VirtualBox:~$ cd /
maxi@maxi-VirtualBox:/$ █
```

4.1.2. **cd**

```
maxi@maxi-VirtualBox:/$ cd
maxi@maxi-VirtualBox:~$ █
```

4.1.3. **cd /etc**

•

```
maxi@maxi-VirtualBox:~$ cd /etc
maxi@maxi-VirtualBox:/etc$
```

4.1.4. cd..

- se detecta como comando “cd..”.

```
maxi@maxi-VirtualBox:~$ cd..
cd..: no se encontró la orden
```

4.1.5. cd ..

```
maxi@maxi-VirtualBox:~$ cd ..
maxi@maxi-VirtualBox:/home$
```

4.2. Bash sobre directorios:

4.2.1. ¿Cuáles son las diferencias entre el **path absoluto** y el **path relativo**?

- Ruta **absoluta** y **relativa** en Linux

La ruta **absoluta** siempre comienza desde el directorio raíz (/). Por ejemplo, /home/indu/scripts/mis_scripts.sh.

• Una ruta **relativa** comienza desde el directorio actual. Por ejemplo, si estás en el directorio /home y querés acceder al archivo my_scripts.sh, podés usar indu/scripts/mis_scripts.sh.

<https://tecnicos.epet1.edu.ar/2021/09/rutas-absolutas-y-relativas-en-linux-cual-es-la-diferencia/>

4.2.2. ¿Qué es lo que realizan las **siguientes** operaciones? (tip: si no encuentra la diferencia primero haga cd /, y luego vuelva a intentar)

```
maxi@maxi-VirtualBox:~$ cd /
```

4.2.2.1. cd ~

```
maxi@maxi-VirtualBox:~$ cd /
maxi@maxi-VirtualBox:$ cd ~
maxi@maxi-VirtualBox:~$
```

4.2.2.2. cd -

•

```
maxi@maxi-VirtualBox:~$ cd -  
/  
maxi@maxi-VirtualBox:/$ cd -  
/home/maxi
```

4.2.3. ¿Cuál es la diferencia entre **cd ~** y **cd -**?

- la diferencia entre **cd ~** y **cd -**, es :

que **cd ~** te manda a la dirección o directorio de **~** = “/home/nombre de usuario, y **cd -** te vuelve a la dirección que estabas anteriormente y muestra la localización del archivo.

4.3. Operaciones con directorios:

4.3.1. ¿Con qué comando se puede **crear** un directorio?

- El comando para poder Crear Carpetas o Directorios es = **mkdir**.

4.3.2. ¿Con qué comando se puede **borrar** un directorio?

- El comando para poder Eliminar Carpetas o Directorios es = **rmdir**.

4.3.3. ¿Qué sucede si el directorio **no** está vacío?

- puede haber subdirectorios y archivos.

TIP= rm -r [directorio] = permite borrar un directorio con subdirectorios y archivos dentro

Si tenemos una carpeta llamada **Dir1** que contiene subcarpetas y archivos, tendremos que añadir el modificador **-r**. El comando se vería así:

```
rm -r Dir1
```

La opción **-r** elimina de forma recursiva las carpetas y su contenido.

https://www.hostinger.com.ar/tutoriales/borrar-archivos-carpetas-linux#El_comando_rm

4.3.4. ¿Cómo puedo salvar la situación anterior? (Sin borrar uno a uno los archivos existentes).

- Para poder borrar un archivo en específico sin borrar los anteriores se debe de utilizar de la siguiente manera el comando **rmdir**.



solo borraría al último **archivo o subdocumento** que está en la ruta que especificaste.

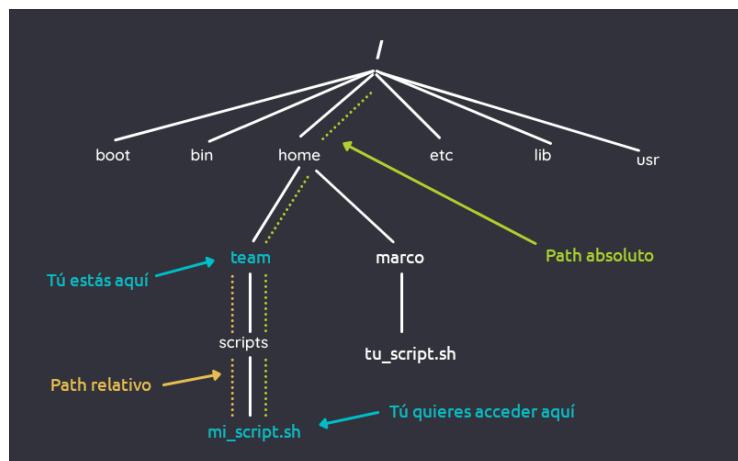
4.4. ¿Qué significa la expresión ./ cuando se utiliza delante de un archivo?

- es para saber donde está ubicado los archivos.

¿Para qué sirve?

- sirve para saber el orden en que se abrieron las expresiones.

que es el PATH = “/”



4.5. ¿Cómo puede moverse entre directorios sin utilizar el PATH completo?

- Se puede utilizar el comando **cd** para poder entrar a distintos directorios y también poder salirse sin utilizar el **PATH**, ejemplo:

```
maxi@maxi-VirtualBox:~$ cd ..  
maxi@maxi-VirtualBox:/home$ █
```

```
maxi@maxi-VirtualBox:/home$ cd ~  
maxi@maxi-VirtualBox:~$ █
```

=====

5. ARCHIVOS

5.1. ¿Qué hacen los siguientes comandos?.

5.1.1. cp

- Lo que te permite hacer el comando **cp** es : “**Copias de Archivos individuales**” o “**de Carpetas Enteras**”.

[https://www.ionos.es/digitalguide/servidores/configuracion/comando-cp-de-linux/#:~:text=Comando%20el%20comando%20CP%20\(abreviatura,individuales%20o%20de%20carpetas%20enteras](https://www.ionos.es/digitalguide/servidores/configuracion/comando-cp-de-linux/#:~:text=Comando%20el%20comando%20CP%20(abreviatura,individuales%20o%20de%20carpetas%20enteras)

5.1.2. mv

- Lo que te permite hacer el comando **mv** es : “**Mover o renombrar archivos o directorios del sistema de archivos**”.

<https://es.wikipedia.org/wiki/Mv#:~:text=mv%20es%20un%20comando%20de,o%20puede%20ser%20el%20mismo.>

5.1.3. rm

- Lo que te permite hacer el comando **rm** es : Es un comando de la familia de sistemas operativos Unix usada para eliminar archivos y directorios del sistema de archivos. Esta orden debe utilizarse con cautela, ya que puede ser muy destructiva, debido a que, al momento de ser llamada, por omisión borra los archivos sin pedir confirmación.

<https://es.wikipedia.org/wiki/Rm#:~:text=rm%20es%20un%20comando%20de,los%20archivos%20sin%20pedir%20confirmaci%C3%B3n.>

5.1.4. rcp

- Lo que te permite hacer el comando **rcp** es : le permite copiar archivos de una sistema a otro. Éste utiliza los archivos /etc/hosts. equiv y /etc/passwd del sistema remoto para determinar si dispone de derechos de acceso al sistema remoto. La sintaxis del comando **rcp** es similar a la usada con **cp** .

<https://docs.oracle.com/cd/E19620-01/805-7644/6j76kloqn/index.html#:~:text=El%20comando%20rcp%20le%20permite,a%20la%20usada%20con%20cp%20.>

5.1.5. rsh

- El comando **rsh** (del shell remoto) **le permite ejecutar un único comando en un sistema remoto sin tener que conectar anteriormente**. Esto le puede ahorrar mucho tiempo cuando sólo quiera hacer una cosa en el sistema remoto. De forma parecida a los comandos rlogin y rcp , rsh usa los archivos /etc/hosts.

[https://docs.oracle.com/cd/E19620-01/805-7644/6j76kloqp/index.html#:~:text=El%20comando%20rsh%20\(del%20shell,cosa%20en%20el%20sistema%20remoto.&text=De%20forma%20parecida%20a%20los,los%20archivos%20%2Fetc%2Fhosts](https://docs.oracle.com/cd/E19620-01/805-7644/6j76kloqp/index.html#:~:text=El%20comando%20rsh%20(del%20shell,cosa%20en%20el%20sistema%20remoto.&text=De%20forma%20parecida%20a%20los,los%20archivos%20%2Fetc%2Fhosts)

5.1.6. scp

- El comando scp copia archivos o directorios entre un sistema local y un sistema remoto, o entre dos sistemas remotos. Puede usar este comando desde un sistema remoto (tras iniciar sesión con el comando ssh) o desde el sistema local. El comando **scp** utiliza **ssh** para la transferencia de datos.

https://docs.oracle.com/cd/E37929_01/html/E36611/remotehowtoaccess-55154.html#:~:text=El%20comando%20scp%20copia%20archivos.para%20la%20transferencia%20de%20datos

5.1.7. ssh

- SSH es un programa que **permite acceder a otro ordenador a través de la red, ejecutar comandos en la máquina remota y mover ficheros entre dos máquinas**. Provee autenticación y comunicaciones seguras sobre canales inseguros. Es un reemplazo de rlogin, rsh y rcp.

https://www3.uji.es/~galdu/ssh_vs_rsh/x165.html#:~:text=SSH%20es%20un%20programa%20que,de%20rlogin%2C%20rsh%20y%20rcp

5.2. ¿Con qué comando puedo concatenar el contenido de dos archivos?

catfile1.txtfile2.txt

El comando cat anterior concatenará los dos archivos (file1.txt y file2.txt) y mostrará el salida en la pantalla. Algunas veces el salida no cabrá dentro de la pantalla. En dicha situación puedes mostrar esos archivos en un archivo nuevo o mostrar el archivo utilizando el comando less.

<https://www.hscripts.com/es/tutoriales/linux-commands/cat.html>

5.2.1. ¿Se puede usar ese comando para otra cosa?

- El comando cat, puede realizarse en múltiples comando que habilita por ejemplo la posibilidad de concatenar múltiples archivos y transferir su salida a otro archivo (catfile1.txtfile2.txt>file3.txt). En el ejemplo anterior la salida se redirige al nuevo archivo file3.txt. El comando cat creará el nuevo archivo file3.txt y guardará la salida concatenando en file3.txt.

5.3. Haga un ls -l /dev

- Ls -L /dev

```
maxi@maxi-VirtualBox:~$ ls -l /dev
total 0
crw-r--r-- 1 root root    10, 235 ago  8 12:38 autofs
drwxr-xr-x 2 root root        460 ago  8 12:50 block
drwxr-xr-x 2 root root        80 ago  8 12:32 bsg
crw----- 1 root root    10, 234 ago  8 12:32 btrfs-control
drwxr-xr-x 3 root root        60 ago  8 12:32 bus
lrwxrwxrwx 1 root root        3 ago  8 12:38 cdrom -> sr0
drwxr-xr-x 2 root root     3700 ago  8 12:32 char
crw--w---- 1 root tty      5,  1 ago  8 12:38 console
lrwxrwxrwx 1 root root       11 ago  8 12:32 core -> /proc/kcore
drwxr-xr-x 5 root root     100 ago  8 12:32 cpu
crw----- 1 root root    10, 124 ago  8 12:38 cpu_dma_latency
crw----- 1 root root    10, 203 ago  8 12:32 cuse
drwxr-xr-x 6 root root     120 ago  8 12:32 disk
drwxr-xr-x 2 root root       60 ago  8 12:32 dma_heap
drwxr-xr-x 3 root root       80 ago  8 12:32 dri
lrwxrwxrwx 1 root root       3 ago  8 12:38 dvd -> sr0
crw----- 1 root root    10, 126 ago  8 12:38 ecryptfs
crw-rw--- 1 root video    29,  0 ago  8 12:38 fb0
lrwxrwxrwx 1 root root       13 ago  8 12:32 fd -> /proc/self/fd
crw-rw--- 1 root root      1,  7 ago  8 12:38 full
crw-rw--- 1 root root    10, 229 ago  8 12:38 fuse
crw----- 1 root root   240,  0 ago  8 12:38 hidraw0
crw----- 1 root root    10, 228 ago  8 12:38 hpet
drwxr-xr-x 2 root root       0 ago  8 12:32 hugepages
crw----- 1 root root    10, 183 ago  8 12:38 hwrng
crw----- 1 root root     89,  0 ago  8 12:38 i2c-0
lrwxrwxrwx 1 root root       12 ago  8 12:32 initctl -> /run/initctl
drwxr-xr-x 4 root root     340 ago  8 12:32 input
crw-r---- 1 root root      1, 11 ago  8 12:38 kmsg
lrwxrwxrwx 1 root root      28 ago  8 12:32 log -> /run/systemd/journal/dev-log
brw-rw--- 1 root disk      7,  0 ago  8 12:38 loop0
brw-rw--- 1 root disk      7,  1 ago  8 12:38 loop1
brw-rw--- 1 root disk      7, 10 ago  8 12:38 loop10
brw-rw--- 1 root disk      7, 11 ago  8 12:38 loop11
brw-rw--- 1 root disk      7, 12 ago  8 12:38 loop12
brw-rw--- 1 root disk      7, 13 ago  8 12:38 loop13
brw-rw--- 1 root disk      7, 14 ago  8 12:39 loop14
brw-rw--- 1 root disk      7, 15 ago  8 12:50 loop15
brw-rw--- 1 root disk      7,  2 ago  8 12:38 loop2
brw-rw--- 1 root disk      7,  3 ago  8 12:38 loop3
brw-rw--- 1 root disk      7,  4 ago  8 12:38 loop4
brw-rw--- 1 root disk      7,  5 ago  8 12:38 loop5
brw-rw--- 1 root disk      7,  6 ago  8 12:38 loop6
```

5.3.1. ¿Qué significa el primer carácter?

- El primer carácter indica el tipo de archivo que es.

<https://www.profesionalreview.com/2017/01/28/permisos-basicos-linux-ubuntu-chmod/>

5.3.2. ¿Cuáles son todos los posibles valores que puede contener ese campo y que significa cada uno?

- d: directorio
- b: archivo de bloque
- c: archivo especial de caracteres
- p: canal
- s: socket
- -: archivo normal

Explicación de los valores que contiene el campo:

https://www.youtube.com/watch?v=KhbM0BQ3_ew

<https://www.profesionalreview.com/2017/01/28/permisos-basicos-linux-ubuntu-chmod/>

5.4. ¿Para qué sirve el comando **touch**? ¿Qué utilidad le encuentra?

- **touch** - crear o actualizar ficheros

la utilidad de **touch** es crea un fichero vacío desde el terminal de linux, si el fichero existe le cambia la fecha y hora de modificación con la actual del sistema o podemos especificar la hora con el parámetro --date

<https://cambiatealinux.com/touch-crear-o-actualizar-ficheros>

=====

6. PERMISOS

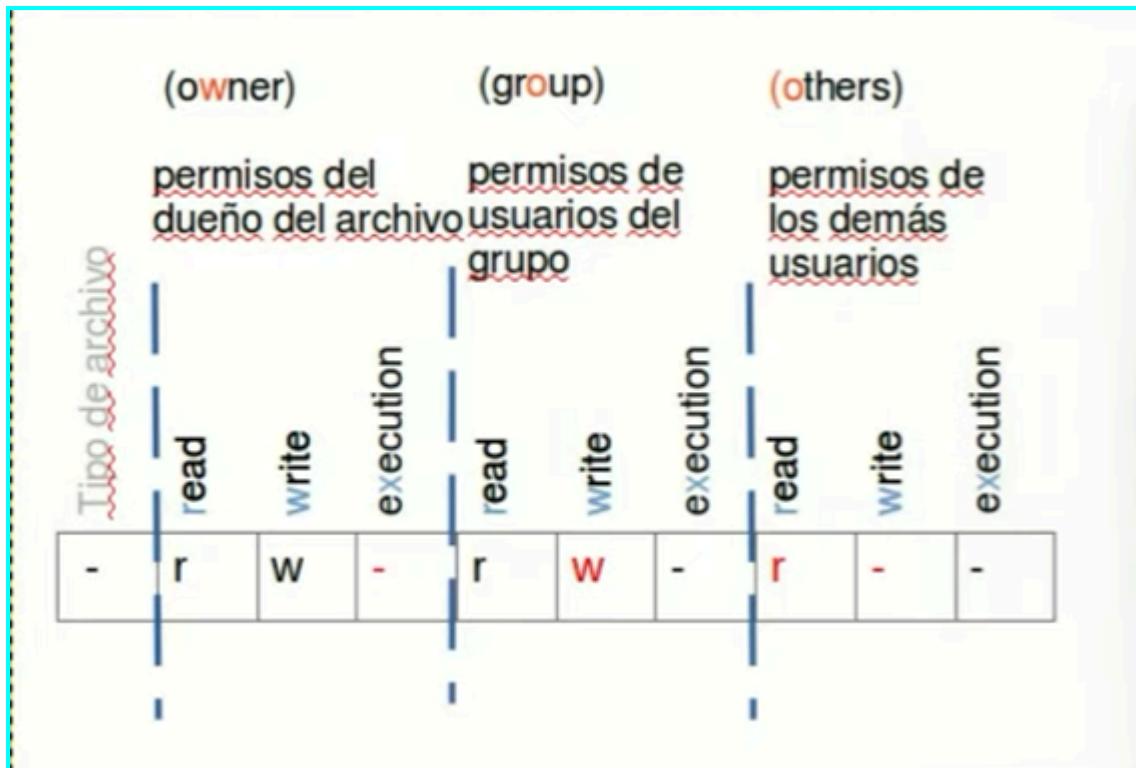
6.1. Teniendo en cuenta el ls –l anterior, ¿indique que son los siguientes 9 caracteres? (sin considerar el primero sobre el que ya respondió anteriormente)

Los siguientes 9 caracteres pueden ser :

- r: significa permiso de lectura (read)
- w: significa permiso de escritura (write)
- x: significa permiso de ejecución (execution)
- -: significa permiso deshabilitado.

<https://www.profesionalreview.com/2017/01/28/permisos-basicos-linux-ubuntu-chmod/>

6.2. ¿qué significa cada carácter? ¿cómo están agrupados?



El primer carácter indica el tipo de archivo que es (se explica de qué tipo pueden ser en el punto 5.3.2).

Los siguientes nueve caracteres se dividen en líneas de 3 caracteres cada una. La primera línea corresponde al propietario del archivo, la segunda línea corresponde a los usuarios del grupo y la tercera es para el resto de usuarios.

<https://www.profesionalreview.com/2017/01/28/permisos-basicos-linux-ubuntu-chmod/>

6.3. ¿Cómo se asignan los permisos? (detalle los comandos).

```
swdev@silver-pc:~/Desktop/carpeta$ sudo groupadd grupo1
```

Primero se crean los grupos a los que les vamos a asignar esos permisos con el sudo groupadd mas el nombre que le queremos asignar..

```
swdev@silver-pc:~/Desktop/carpeta$ sudo adduser elena grupo1
```

Luego se guardan los usuarios que pertenecen a ese grupo mediante el comando sudo su adduser mas el nombre de usuario mas el nombre del grupo.

```
-rwx-rw-r-- 1 pepe pepe 0 Jun 5 14:53 archivo1.txt
-rw-r--r-- 1 pepe pepe 8980 Jun 5 14:47 examples.desktop
p
pepe@silver-pc:~$ chgrp grupo1 archivo1.txt
```

Ahora una vez creado tanto los usuarios y los grupos mediante el comando chgrp mas nombre de grupo y nombre del archivo se le asignan los mismos permisos que tiene el dueño del archivo

```
pepe@silver-pc:~$ chgrp grupo1 archivo1.txt
pepe@silver-pc:~$ ls -l
total 12
-rw-rw-r-- 1 pepe grupo1    0 Jun  5 14:53 archivo1.txt
```

<https://www.ionos.es/digitalguide/servidores/know-how/asignacion-de-permisos-de-acceso-con-chmod/>

6.4. ¿Qué son el **owner**, y el **group** de un archivo? ¿Se pueden cambiar?

- **owner**: Es el usuario que creó y posee un archivo/directorio.
- **group**: Todos los usuarios que son miembros del mismo grupo.

<https://www.hostinger.com.ar/tutoriales/cambiar-permisos-y-propietarios-linux-linea-de-comandos/>

¿Se pueden cambiar?

- Si, se pueden cambiar dependiendo de lo que desee el propietario del archivo o directorio.

6.5. Intente cambiar los permisos de un archivo perteneciente al **root** (sesionado como **usuario**). Explique qué sucedió.

Cuando un usuario sin los permisos para modificar el archivo intenta modificarlos salta un mensaje de permiso denegado.

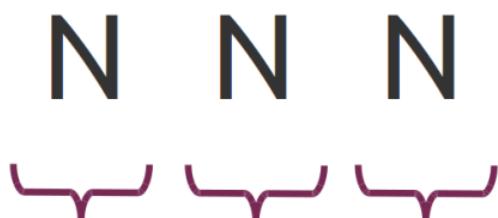
```
maria@silver-pc:/home/pepe$ echo kfljsdfjlfadsjf > archivo1.txt
bash: archivo1.txt: Permission denied
```

hace lo que hace en el video:

https://www.youtube.com/watch?v=KhbM0BQ3_ew

6.6. Explique la forma de cambiar los permisos con valores en octal.

- La forma de cambiar los permisos con valores en **octal** es, sustituyendo cada tipo de permiso (r, w y x) por un 1 o 0, dependiendo de si se da o no el permiso.



Permisos de Usuario

Permisos de Grupo

Permisos de Otros

Número Octal que representa la combinación de permisos de lectura, escritura y ejecución para el propietario

Número Octal que representa la combinación de permisos de lectura, escritura y ejecución para el grupo

Número Octal que representa la combinación de permisos de lectura, escritura y ejecución para el resto de usuarios

Al final, del número en binario obtenido, el número decimal que le corresponde es el que se utiliza para representar las diferentes combinaciones rwx en cada grupo o nivel de permisos (usuario, grupo y otros).

rwx		111		7
rw-		110		6
r-x		101		5
r--		100		4
-wx		011		3
-w-		010		2
--x		001		1
---		000		0

<https://computernewage.com/2015/06/27/conoce-la-estructura-de-permisos-de-linux-al-detalle/#representacion-permisos>

6.7. ¿Cuál es el significado de los permisos en los directorios (se debe indicar que indica una r, una w, y una x)?

- rwxrwxrwx



Especial

Permisos de Usuario

Permisos de Grupo

Permisos de Otros

Tipo de archivo

Permisos de lectura, escritura y ejecución para el propietario

Permisos de lectura, escritura y ejecución para el grupo

Permisos de lectura, escritura y ejecución para el resto de usuarios

a continuación representan los permisos del propietario del fichero o directorio, los del grupo al que pertenece este usuario, y los del resto de usuarios. Aquí tienes el significado:

- r | Se dan permisos de lectura
- w | Se dan permisos de escritura
- x | Se dan permisos de ejecución

<https://computernewage.com/2015/06/27/conoce-la-estructura-de-permisos-de-linux-al-detalle/#representacion-permisos>

7. FILTROS

7.1. ¿Cuál es la diferencia de los comandos **more**, **less** y **cat**? De un ejemplo de cada uno.

- la diferencias entre los comandos **cat**, **more** y **less** son :

•Comando Unix CAT=

El comando ‘cat’ imprimirá por pantalla el contenido del fichero sin ningún tipo de paginación ni posibilidad de modificarlo. Básicamente concatena archivos o la salida estándar en la salida estándar. Podemos pasarle parámetros como:

```
-A, --show-all      lo mismo que -vET
-b, --number-nonblank  numera las líneas que no están vacías
-e                lo mismo que -vE
-E, --show-ends    muestra un $ al final de cada línea
-n, --number       numera todas las líneas
-s, --squeeze-blank nunca muestra más de una línea vacía,
-t                equivalente a -vT
-T, --show-tabs    muestra los caracteres de tabulación como ^I
-u                (sin efecto)
-v, --show-nonprinting  utiliza la notación ^ y M-, salvo para LFD y TAB
--help            muestra esta ayuda y finaliza
--version         informa de la versión y finaliza
```

•Comando Unix MORE=

Al igual que ‘cat’, ‘more’ permite visualizar por pantalla el contenido de un fichero de texto, con la diferencia con el anterior de que ‘more’ pagina los resultados. Primero mostrará por pantalla todo lo que se pueda visualizar sin hacer scroll y después, pulsando la tecla espacio avanzará de igual modo por el fichero.

```
$ more fichero
texto de ejemplo
texto de ejemplo
texto de ejemplo
--Más--(23%)
```

•Comando Unix LESS=

El comando ‘less’ es el más completo de los tres, pues puede hacer todo lo que hace ‘more’ añadiendo mayor capacidad de navegación por el fichero (avanzar y retroceder) además de

que sus comandos están basados en el editor ‘vi’, del cual se diferencia en que no tiene que leer todo el contenido del fichero antes de ser abierto. Tiene una gran cantidad de opciones y parámetros, como siempre lo recomendable:

```
$ less --help
```

<https://rm-rf.es/unix-diferencias-entre-los-comandos-cat-more-y-less/>

7.2. ¿Cuál es la diferencia entre **tail** y **head**?

- **tail**: Muestra las últimas 10 líneas de un archivo específico. Con la opción “-n” puedo modificar la cantidad de líneas que veo. Con la opción -f puedo poner los cambios en escucha
- **head**: Muestra las primeras 10 líneas de un archivo específico. Con la opción “-n” puedo modificar la cantidad de líneas que veo.

<https://platzi.com/clases/1667-linux/22993-diferencias-entre-less-cat-head-y-tail-para-lectura#:~:text=Tail%3A%20muestra%20la%20ultima%20parte.parametros%20de%20lineas%20a%20ver.>

7.3. ¿Qué es lo que realiza el comando **uniq**?

- El comando **uniq** en Linux es una utilidad de línea de comando que informa o filtra las líneas repetidas en un archivo. En palabras simples, **uniq** es la herramienta que ayuda a detectar las líneas duplicadas adyacentes y también elimina las líneas duplicadas.

https://www.compuhoy.com/como-se-usa-uniq-linux/#%C2%BFComo_usar_el_comando_Uiq_en_Linux

7.4. ¿Qué es lo que realiza el comando **grep**?

- **Grep** (por sus siglas en inglés Globally Search For Regular Expression and Print out) es una herramienta de líneas de comando usada para buscar cadenas de texto y encontrar coincidencias dentro de este. También puede ser utilizada para encontrar una palabra o combinación de palabras dentro de un fichero.

<https://keepcoding.io/blog/que-es-el-comando-grep-y-como-usarlo-en-linux/>

8. VI

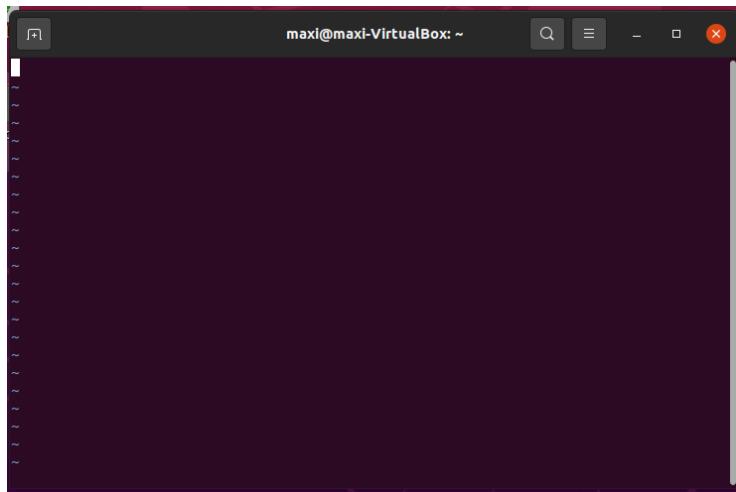
TIP= Cómo se utiliza el Comando **VI**:

<https://www.unirioja.es/cu/enriquez/docencia/Quimica/vi.pdf>

8.1. Ejecute la siguiente instrucción: **vi \$HOME/prueba.txt** ¿Qué sucedió? Ahora ejecute

todos los pasos detallados a continuación.

- `maxi@maxi-VirtualBox:~$ vi $HOME/prueba.txt`

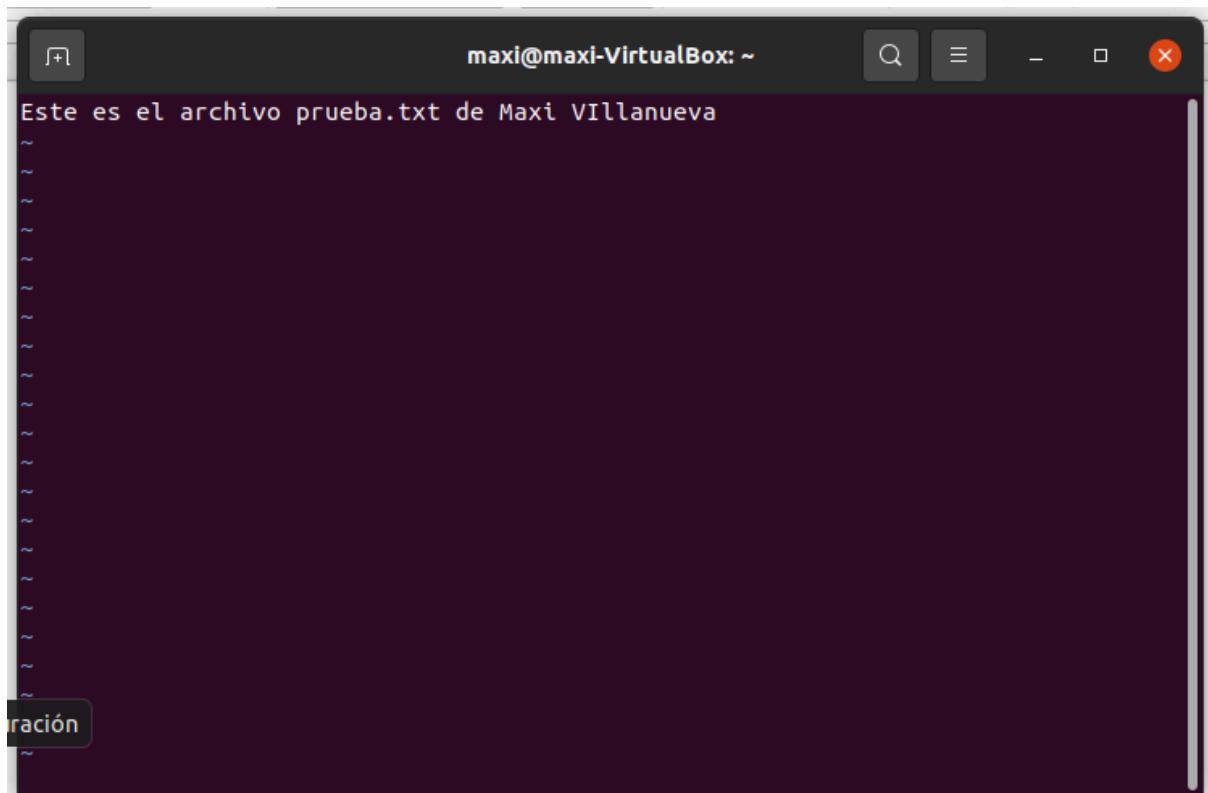


A screenshot of a terminal window titled "maxi@maxi-VirtualBox: ~". The window contains the command "vi \$HOME/prueba.txt" followed by a cursor. The background of the terminal is dark, and the text is white.

Lo que hace el comando **VI** es crear un archivo texto o tipo “.txt” que permite escribir dentro de la terminal.

8.1.1. Escriba la siguiente frase: “**Este es el archivo prueba.txt de <nombre y apellido>**”

-



A screenshot of a terminal window titled "maxi@maxi-VirtualBox: ~". The window contains the text "Este es el archivo prueba.txt de Maxi Villanueva" followed by a cursor. The background of the terminal is dark, and the text is white. A small tooltip-like box labeled "Escritura" is visible at the bottom left of the screen.

8.1.2. ¿Qué tuvo que hacer para poder escribir la frase?

Cambio de modo comando a texto:

Comando	Acción
i	Inserta texto a la izquierda del cursor
a	Inserta texto a la derecha del cursor
A	Inserta texto al final de la línea donde se encuentra el cursor
I	Inserta texto al comienzo de la línea donde se encuentra el cursor
o	Abre una línea debajo de la actual
O	Abre una línea encima de la actual

La acción que hice fue apretar la tecla “a” y luego ingresar el texto

8.1.3. Guarde el archivo, y salga del editor. ¿Qué comando utilizó?

:wq	Guardar cambios y salir
-----	-------------------------

8.1.4. Ingrese nuevamente al archivo.

- para poder ingresar nuevamente al archivo se tiene que volver a ingresar el comando = “vi” y a continuación la posición del archivo en este caso “\$HOME/prueba.txt”.

```
maxi@maxi-VirtualBox:~$ vi $HOME/prueba.txt
```

y apretar la tecla ENTER.

8.1.5. Incorpore al inicio del archivo el siguiente párrafo (los acentos pueden ser evitados):

“Sistemas Operativos

Curso de los días <día de cursada>

Trabajo Práctico 1

Alumno: <su nombre y apellido aquí>

Legajo: <su Legajo aquí>

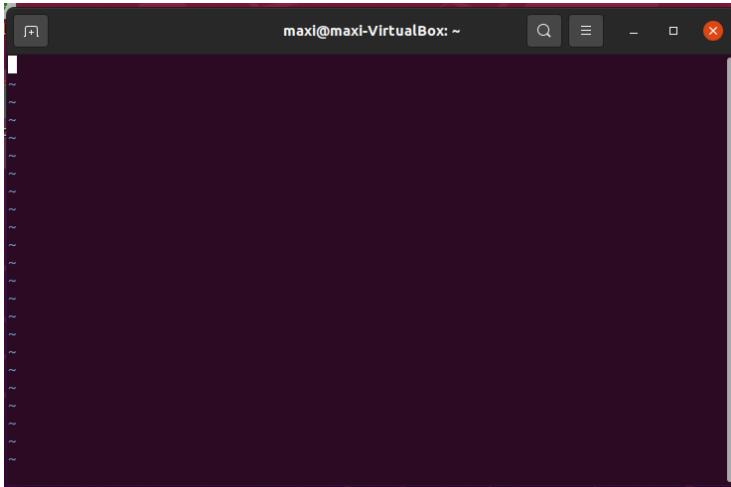
Documento: <su documento aquí>”

•

```
Este es el archivo prueba.txt de Maxi Villanueva
Sistemas Operativos
Trabajo Practico 1
Curso de los dias 10
Alumno: Maximiliano Villanueva
Legajo: 26994
Documento: 44522393
```

8.1.6. Describa todos los pasos que tuvo que realizar.

- Primer paso abrir la terminal y ejecutar el siguiente comando = “vi \$HOME/prueba.txt”
- Segundo Paso una vez ingresado el comando se cargará en el terminal para poder lograr escribir dentro de ella.



Para poder escribir hay que presionar la tecla “a”.

- Tercer paso hacer lo que te indica el punto “8.1.5”.

```
Este es el archivo prueba.txt de Maxi Villanueva
Sistemas Operativos
Trabajo Practico 1
Curso de los dias 10
Alumno: Maximiliano Villanueva
Legajo: 26994
Documento: 44522393
```

Una vez ingresado cada dato presionas la tecla “Esc” para luego ingresar el siguiente comando :

:wq	Guardar cambios y salir
-----	-------------------------

8.1.7. Guarde el archivo y continúe la edición. ¿Qué comandos utilizó?

- | | |
|----|-----------------|
| :w | Guardar cambios |
|----|-----------------|

8.1.8. Borre la línea de “Legajo”. Indique por lo menos dos formas de realizarlo.

-

Borrar texto:

Comando	Acción
x	Borra el carácter bajo el cursor
dd	Borra la línea donde se encuentra el cursor
ndd	Borra las próximas n líneas
D	Borra desde donde se encuentra el cursor hasta el final de la línea
dw	Borra desde donde se encuentra el cursor hasta el final de una palabra

8.1.9. Invierte el orden de las líneas “**Curso** y **TP**”. No está permitido rescribirlas. ¿Qué comandos utilizó?

- estado actual del texto:

This screenshot shows a terminal window titled "maxi@maxi-VirtualBox: ~". The terminal displays the following text:
Este es el archivo prueba.txt de Maxi Villanueva
Sistemas Operativos
Curso de los días 10
Trabajo Práctico 1
Alumno: Maximiliano Villanueva
Legajo: 26994
Documento: 44522393

los 3 comandos que utilice para poder lograr invertir el texto son =

Y	Copia la línea actual
p	Pega después del cursor
P	Pega antes del cursor

se puede usar “P” o “p”, sólo cambia la manera en que se pega el texto

dd	Borra la línea donde se encuentra el cursor
----	---

y quedaría el texto en la terminal así =

This screenshot shows a terminal window titled "maxi@maxi-VirtualBox: ~". The terminal displays the following text:
Este es el archivo prueba.txt de Maxi Villanueva
Sistemas Operativos
Trabajo Práctico 1
Curso de los días 10
Alumno: Maximiliano Villanueva
Legajo: 26994
Documento: 44522393

https://www.linuxtotal.com.mx/index.php?cont=info_admon_010

8.1.10. Ubíquese en la línea 2 (dos) del archivo. No está permitido usar las teclas del cursor, ni el mouse. ¿Qué comando utilizó?

- para poder ubicarse en la linea 2(dos) del archivo se debe de ingresar el siguiente comando= “:2”

8.1.11. Marque para copiar las líneas 2, 3, y 4 (todas juntas, no de a una a la vez). ¿Cómo lo realizó?

- 1. Coloque el cursor donde desea comenzar a cortar.
 2. Presione v para seleccionar caracteres (o V mayúscula para seleccionar líneas completas).
 3. Mueva el cursor al final de lo que desea cortar.
- Utilizar el comando **yy**, cuya función es copiar la línea donde se encuentra situado el cursor.



- Pegar el texto con el comando **p**.

A screenshot of a terminal window showing the result of pasting the copied lines. The text now includes:

```
Este es el archivo prueba.txt de Maxi Villanueva
Sistemas Operativos
Curso de los dias 10
Trabajo Practico 1
Alumno: Maximiliano Villanueva
Legajo: 26994
Documento: 44522393
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
3 lines yanked
```

The lines from "Sistemas Operativos" to "Trabajo Practico 1" have been repeated three times below the original text, indicating they were copied and pasted.

https://www.compuhoy.com/como-se-copian-varias-lineas-en-linux/#%C2%BFComo_se_copian_y_egan_varias_lineas_en_vi

https://www.linuxtotal.com.mx/index.php?cont=info_admon_010

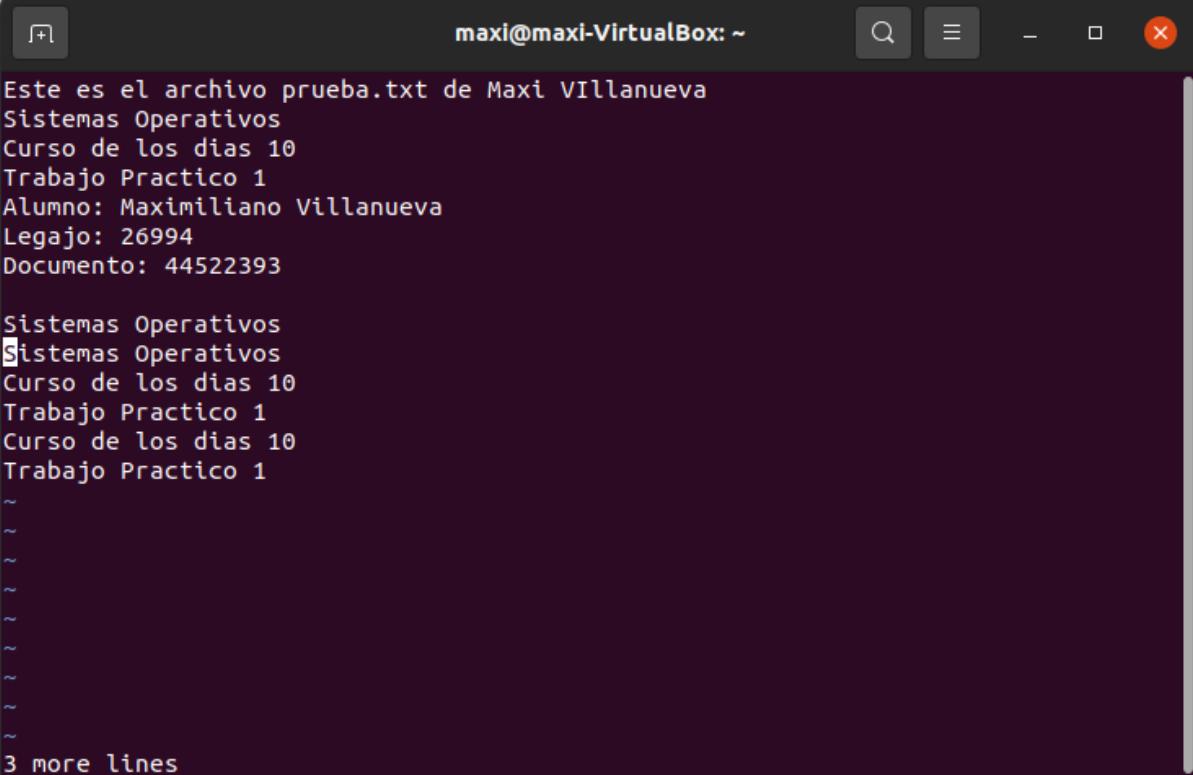
<https://www.unirioja.es/cu/enriquez/docencia/Quimica/vi.pdf>

8.1.12. Ubíquese al final del archivo (sin usar las teclas del cursor), y pegue dos veces el contenido del **buffer**. ¿Qué comando usó?

- El primer comando que utilice fue el “:8” que me permite ingresar la posición que esté el cursor.

También para estar la última línea se puede utilizar el comando **Shift+g**.

- Pegar el texto con el comando **p**.



This screenshot shows a terminal window with the title bar "maxi@maxi-VirtualBox: ~". The terminal displays the following text:

```
Este es el archivo prueba.txt de Maxi Villanueva
Sistemas Operativos
Curso de los dias 10
Trabajo Practico 1
Alumno: Maximiliano Villanueva
Legajo: 26994
Documento: 44522393

Sistemas Operativos
Sistemas Operativos
Curso de los dias 10
Trabajo Practico 1
Curso de los dias 10
Trabajo Practico 1

~
~
~
~
~
~
~
3 more lines
```

<https://www.unirioja.es/cu/enriquez/docencia/Quimica/vi.pdf>

8.1.13. Deshaga uno de los copiados. No está permitido borrar línea por línea, ni carácter a carácter. ¿Qué comando usó?

•

VI Editing commands

1. i – Insert at cursor (goes into insert mode)
2. a – Write after cursor (goes into insert mode)
3. A – Write at the end of line (goes into insert mode)
4. ESC – Terminate insert mode.
5. u – Undo last change.
6. U – Undo all changes to the entire line.
7. o – Open a new line (goes into insert mode)
8. dd – Delete line.

Más elementos... • 25 jun 2022

<https://www.guru99.com/the-vi-editor>

[VI Editor with Commands in Linux/Unix Tutorial - Guru99](https://www.guru99.com/the-vi-editor)

La tecla **u** te permite volver atrás, de esa manera te evitas borrar las líneas

```
Este es el archivo prueba.txt de Maxi Villanueva
Sistemas Operativos
Curso de los dias 10
Trabajo Practico 1
Alumno: Maximiliano Villanueva
Legajo: 26994
Documento: 44522393

Sistemas Operativos
Sistemas Operativos
Curso de los dias 10
Trabajo Practico 1
Curso de los dias 10
Trabajo Practico 1
~
~
~
~
~
~
~
3 more lines
```

```
Este es el archivo prueba.txt de Maxi Villanueva
Sistemas Operativos
Curso de los dias 10
Trabajo Practico 1
Alumno: Maximiliano Villanueva
Legajo: 26994
Documento: 44522393

~
~
~
~
~
~
~
~
~
~
~
~
~
5 fewer lines; after #4 15 seconds ago
```

8.1.14. ¿Cómo busco la palabra “Documento”? ¿Cómo busco la segunda ocurrencia de una palabra?

- /Documento = busca desde la palabra de arriba, y luego apretar “N” busca la palabra de **abajo** si ahí otra igual y si apretas “n” busca la palabra de **arriba** si ahí otra igual.
 - ?Documento = busca desde la palabra de abajo, y luego apretar “N” busca la palabra de **arriba** si ahí otra igual y si apretas “n” busca la palabra de **abajo** si ahí otra igual.

8.1.15. ¿Cómo puedo reemplazar la palabra “Documento” por “Documento:” (sin borrar, o realizar el reemplazo a mano)

- `:%s/Documento/Documento:/gc`

% = para todos

s = sustituir

g = que busque todos los Documento y los reemplace

c = que confirme todas las sustituciones y las reemplace

y=yes

n=no

si quieras reemplazar la palabra poner y, si no quieres ingresar n

```
Este es el archivo prueba.txt de Maxi Villanueva
Sistemas Operativos
Curso de los dias 10
Trabajo Practico 1
Alumno: Maximiliano Villanueva
Legajo: 26994
Documento: 44522393

~
```

8.1.15. link=

<https://linuxize.com/post/vim-find-replace/#:~:text=quick%20and%20easy.-,Basic%20Find%20and%20Replace,press%20the%20%27Esc%27%20key>

8.1.16. Guarde el archivo y salga.

- 1

8.1.17. Ejecutar "vi miprog.c"

A screenshot of a terminal window titled "maxi@maxi-VirtualBox: ~". The window contains a single line of text: "miprogr.c" [New File]. The terminal interface includes a search bar, a menu icon, and standard window control buttons (minimize, maximize, close) in the top right corner.

```
miprogr.c [New File]
```

introducir el texto:

```
principal ()  
{  
printf (" 1/3/2001 ----- listo");  
printf (" 1/3/2002 ----- listo");  
printf (" 1/3/2003 ----- listo");  
printf (" 1/3/2004 ----- listo");  
printf (" 1/3/2005 ----- listo");  
printf (" 1/3/2006 ----- No listo");  
}
```

```
principal()  
{  
printf(" 1/3/2001 ----- listo");  
printf(" 1/3/2002 ----- listo");  
printf(" 1/3/2003 ----- listo");  
printf(" 1/3/2004 ----- listo");  
printf(" 1/3/2005 ----- listo");  
printf(" 1/3/2006 ----- No listo");  
}█
```

Luego presionar :x

- o Escribir ":x"

ahora ejecute los siguientes comandos:

gcc miprog.c -o miprimerprog (si no tiene instalado **gcc**, hágalo usando el comando **sudo apt install gcc**).

```
maxi@maxi-VirtualBox:~$ gcc miprog.c -o miprimerprog  
miprog.c:1:1: warning: return type defaults to 'int' [-Wimplicit-int]  
1 | principal()  
| ^~~~~~  
miprog.c: In function 'principal':  
miprog.c:3:1: warning: implicit declaration of function 'printf' [-Wimplicit-function-declaration]  
3 | printf(" 1/3/2001 ----- listo");  
| ^~~~~~  
miprog.c:3:1: warning: incompatible implicit declaration of built-in function 'printf'  
miprog.c:1:1: note: include '<stdio.h>' or provide a declaration of 'printf'  
+++ |+#include <stdio.h>  
1 | principal()  
/usr/bin/ld: /usr/lib/gcc/x86_64-linux-gnu/9/../../../../x86_64-linux-gnu/Scrt1.o: en la función `__start':  
.text+0x24: referencia a `main' sin definir  
collect2: error: ld returned 1 exit status  
maxi@maxi-VirtualBox:~$ █
```

./miprimerprog

¿Cuál es la salida?

volver a ejecutar **vi miprog.c**

poner "%s/l3/lMarzo/g" ¿Que pasó al ejecutar esto?

los errores que que corregir fueron los siguientes =

```
#include<stdio.h>

int main()
{
printf(" 1/Marzo/2001 ----- listo");
printf(" 1/Marzo/2002 ----- listo");
printf(" 1/Marzo/2003 ----- listo");
printf(" 1/Marzo/2004 ----- listo");
printf(" 1/Marzo/2005 ----- listo");
printf(" 1/Marzo/2006 ----- No listo \n");

return 0;
}
```

Corrija los errores que surgieron al utilizar el comando `acc`.

volver a ejecutar :x

ahora volver a ejecutar:

```
gcc miprog.c -o miprimerprog  
./miprimerprog
```

¿Cuál es la salida ahora?

```
maxi@maxi-VirtualBox:~$ vi miprog.c  
maxi@maxi-VirtualBox:~$ gcc miprog.c -o miprimerprog  
maxi@maxi-VirtualBox:~$ ./miprimerprog  
1/Marzo/2001 ----- listo 1/Marzo/2002 ----- listo 1/Marzo/2003 ----- listo 1/Marzo/2004 ----- listo 1/Marzo/2005 ----- listo 1/Marzo/2006 ----- No listo  
maxi@maxi-VirtualBox:~$
```

9. VARIABLES DE ENTORNO

9.1. ¿Qué son las variables de entorno y para qué sirven?

- Las **variables de entorno** son **variables** definidas para el shell actual y heredadas por cualquier shell o proceso secundario. Las variables de entorno se utilizan para transmitir información a procesos que se producen desde el shell.

<https://www.digitalocean.com/community/tutorials/how-to-read-and-set-environmental-and-shell-variables-on-linux-es#:~:text=Las%20variables%20de%20entorno%20son,se%20producen%20desde%20el%20shell>.

9.1.1. **Escriba el contenido** y explique el significado de las siguientes variables: HOME / LOGNAME / PATH / HOSTNAME / IFS

- el contenido de las variables =

```
maxi@maxi-VirtualBox:~$ echo $HOME  
/home/maxi  
maxi@maxi-VirtualBox:~$ echo $LOGNAME  
maxi  
maxi@maxi-VirtualBox:~$ echo $PATH  
/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games:/snap/bin  
maxi@maxi-VirtualBox:~$ echo $HOSTNAME  
maxi-VirtualBox  
maxi@maxi-VirtualBox:~$ echo $IFS  
maxi@maxi-VirtualBox:~$
```

- significado de las siguientes variables=

Home= Es el directorio de los usuarios estándar, y por lo tanto, el destinado a almacenar todos los archivos del usuario, como documentos, fotos, vídeos, música, plantillas, etc.

LOGNAME = comando muestra el nombre de usuario. la ejecución de comandos logname, se mostrará el nombre del usuario actual.

El **PATH** = informa al shell (en la mayoría de los casos BASH) dónde se encuentran los programas binarios que puede ejecutar en el sistema, sin tener que llamarlos por su ruta absoluta).

El comando **hostname** se implementa y ha sido desarrollada para mostrar o establecer el nombre de host y el nombre de dominio de un equipo en particular y es uno de los comandos.

La variable **IFS** se usa en shells (Bourne, POSIX, ksh, bash) como separador de campo de entrada (o separador de campo interno). Esencialmente, es una cadena de caracteres especiales que deben tratarse como delimitadores entre palabras / campos al dividir una línea de entrada. El valor predeterminado de IFS es espacio, tabulación, nueva línea.

<https://www.youtube.com/watch?v=E3btf4iUGpA>

<https://computernewage.com/2015/06/14/el-arbol-de-directorios-de-linux-al-detalle-que-contiene-cada-carpeta/#:~:text=deber%C3%A0contener%20binarios.,Home.las%20configuraciones%20de%20programas%2C%20etc>

<http://www.w3big.com/es/linux/linux-comm-username.html#:~:text=Linux%20LOGNAME%20comando%20muestra%20el.el%20nombre%20del%20usuario%20actual.>

[https://rootsudo.wordpress.com/2014/04/06/el-path-la-ruta-de-linux-variables-de-entorno/#:~:text=El%20PATH%20informa%20al%20shell.llamarlos%20por%20su%20ruta%20absoluta\).&text=vemos%20que%20separados%20por%20dos,binarios%20\(en%20ese%20orden\).](https://rootsudo.wordpress.com/2014/04/06/el-path-la-ruta-de-linux-variables-de-entorno/#:~:text=El%20PATH%20informa%20al%20shell.llamarlos%20por%20su%20ruta%20absoluta).&text=vemos%20que%20separados%20por%20dos,binarios%20(en%20ese%20orden).)

<https://www.solvetic.com/tutoriales/article/5109-como-usar-comando-hostname-en-linux/#:~:text=El%20comando%20hostname%20se%20implementa,es%20uno%20de%20los%20comandos.>

https://www.linuxteaching.com/article/how_to_use_ifs_in_bash#what_is_ifs_in_bash

9.1.2. ¿Qué comando usó para ver el contenido de las variables del punto anterior?

- Para ver el contenido de las variables utilice la variable **echo**.

9.1.3. Cree una variable de entorno **HOLA** que contenga el mensaje “Hola mundo”.

- para poder crear una variable de entorno se utiliza comando **export** y para crear **HOLA** y nos muestre “Hola mundo” se hace así:

```
maxi@maxi-VirtualBox:~$ export HOLA="hola mundo"
```

y para poder mostrar la variable y nos muestre su contenido se con la variable **echo** y se utiliza de esta manera:

```
maxi@maxi-VirtualBox:~$ echo $HOLA  
hola mundo
```

<https://www.youtube.com/watch?v=VSmfKeTkL48>

9.1.4. ¿Cuál es el uso que le da el sistema a la variable **PATH**?

- **PATH** es la variable del sistema que utiliza el sistema operativo para buscar los ejecutables necesarios desde la línea de comandos o la ventana Terminal.

¿Qué ocurre si intenta ejecutar un comando que no se encuentra ubicado en alguno de los directorios que contiene la variable?

- Al momento de invocar un **echo** con una variable que no se encuentra ubicado en ningun directorio, aparece un espacio en blanco.

¿Cómo lo soluciona?

- Creando una variable de entorno, con una línea de texto para poder, luego, poder plasmarlo en la consola.

9.1.5. ¿Por qué existen las variables PS1 y PS2?

- **PS1**: variable de entorno que contiene el valor del indicador predeterminado. Cambia la apariencia y el entorno de la línea de comandos de la shell.

• **PS2**: variable de entorno que contiene el valor que el indicador utilizó para una interpretación de continuación del comando. Lo ve cuando escribe un comando largo en muchas líneas.

¿Qué es un comando multilínea?

- Los comandos de múltiples líneas requieren el uso de operadores de asignación. Esto significa que el resultado de cualquier función, expresión u otra operación sea asignada a una variable.

https://help.seagullscientific.com/2021/es/subsystems/btvbscript/content/sib_multi_line_scripts.htm

BIBLIOGRAFÍA

- 2.2 link=

<https://cybersophia.net/linux/etc-passwd-file-in-linux/#:~:text=To%20display%20the%20content%20of,in%20Linux%2C%20as%20described%20below.&text=To%20filter%20the%20output%20of,in%20Linux%2C%20as%20shown%20below>

- 2.3.6. link = <https://tecnoysoft.com/es/terminal-tty-en-gnulinux/>
-

- 3.2.1. link =

<https://unix.stackexchange.com/questions/19451/difference-between-help-info-and-man-command>

- 3.3.2. link =

https://www.profesionalreview.com/2016/07/23/ayuda-comandos-linux/#:~:text=help,_opciones%20de%20un%20determinado%20comando.

- 3.3.2. link =

<https://askubuntu.com/questions/832562/difference-among-whereis-locate-and-find-command>

- 4.2.1 link =

<https://tecnicos.epet1.edu.ar/2021/09/rutas-absolutas-y-relativas-en-linux-cual-es-la-diferencia/>

- 4.3.3 link=

https://www.hostinger.com.ar/tutoriales/borrar-archivos-carpetas-linux#El_comando_rm

- 5.1.1 link=

[https://www.ionos.es/digitalguide/servidores/configuracion/comando-cp-de-linux/#:~:text=Comando%20el%20comando%20CP%20\(abreviatura,individuales%20o%20de%20carpetas%20enteras](https://www.ionos.es/digitalguide/servidores/configuracion/comando-cp-de-linux/#:~:text=Comando%20el%20comando%20CP%20(abreviatura,individuales%20o%20de%20carpetas%20enteras)

=====

5.1.2. link=

<https://es.wikipedia.org/wiki/Mv#:~:text=mv%20es%20un%20comando%20de,o%20puede%20ser%20el%20mismo.>

=====

- 5.1.3 link=

<https://es.wikipedia.org/wiki/Rm#:~:text=rm%20es%20un%20comando%20de,los%20archivos%20sin%20pedir%20confirmaci%C3%B3n.>

=====

- 5.1.4 link=

<https://docs.oracle.com/cd/E19620-01/805-7644/6j76kloqn/index.html#:~:text=El%20comando%20rcp%20le%20permite,a%20la%20usada%20con%20cp%20.>

=====

- 5.1.5 link =

[https://docs.oracle.com/cd/E19620-01/805-7644/6j76kloq/index.html#:~:text=El%20 comando%20rsh%20\(del%20shell,cosa%20en%20el%20sistema%20remoto.&text=De%20forma%20parecida%20a%20los,los%20archivos%20%2Fetc%2Fhosts.](https://docs.oracle.com/cd/E19620-01/805-7644/6j76kloq/index.html#:~:text=El%20 comando%20rsh%20(del%20shell,cosa%20en%20el%20sistema%20remoto.&text=De%20forma%20parecida%20a%20los,los%20archivos%20%2Fetc%2Fhosts.)

=====

- 5.2. link =

<https://www.hscripts.com/es/tutoriales/linux-commands/cat.html>

=====

- 5.4. link =

<https://cambiatealinux.com/touch-crear-o-actualizar-ficheros>

=====

- 5.3. link =

<https://www.profesionalreview.com/2017/01/28/permisos-basicos-linux-ubuntu-chmod/>

=====

- 5.3.1. link =

https://www.youtube.com/watch?v=KhbM0BQ3_ew

<https://www.profesionalreview.com/2017/01/28/permisos-basicos-linux-ubuntu-chmod/>

=====

- 5.3.2 link =

https://www.youtube.com/watch?v=KhbM0BQ3_ew

<https://www.profesionalreview.com/2017/01/28/permisos-basicos-linux-ubuntu-chmod/>

- 6.1 link=

<https://www.profesionalreview.com/2017/01/28/permisos-basicos-linux-ubuntu-chmod/>

- 6.2 link=

<https://www.profesionalreview.com/2017/01/28/permisos-basicos-linux-ubuntu-chmod/>

- 6.3. link =

<https://docs.oracle.com/cd/E19620-01/805-7644/x-5lbb/index.html>

- 6.4 link=

<https://www.hostinger.com.ar/tutoriales/cambiar-permisos-y-propietarios-linux-linea-de-comandos/>

- 6.5 link=

https://www.youtube.com/watch?v=KhbM0BQ3_ew

- 6.6 link=

<https://computernewage.com/2015/06/27/conoce-la-estructura-de-permisos-de-linux-al-detalle/#representacion-permisos>

- 6.7 link=

<https://computernewage.com/2015/06/27/conoce-la-estructura-de-permisos-de-linux-al-detalle/#representacion-permisos>

- 7.1. link=

<https://rm-rf.es/unix-diferencias-entre-los-comandos-cat-more-y-less/>

- 7.2. link=

<https://platzi.com/clases/1667-linux/22993-diferencias-entre-less-cat-head-y-tail-para-lectura#:~:text=Tail%3A%20muestra%20la%20ultima%20parte,parametros%20de%20lineas%20a%20ver.>

- 7.3. link=

https://www.compuhoy.com/como-se-usa-uniq-linux/#%C2%BFComo_usar_el_comando_Uniq_en_Linux

- 7.4 link=

<https://keepcoding.io/blog/que-es-el-comando-grep-y-como-usarlo-en-linux/>

- VI link=

<https://www.unirioja.es/cu/enriquez/docencia/Quimica/vi.pdf>

- 8.1.9. link=

https://www.linuxtotal.com.mx/index.php?cont=info_admon_010

- 8.1.11 Link 1 =

https://www.compuhoy.com/como-se-copian-varias-lineas-en-linux/#%C2%BFComo_se_copian_y_pegan_varias_lineas_en_vi

- 8.1.11 Link 2 =

https://www.linuxtotal.com.mx/index.php?cont=info_admon_010

- 8.1.11 Link 3 =

<https://www.unirioja.es/cu/enriquez/docencia/Quimica/vi.pdf>

- 8.1.12. link=

<https://www.unirioja.es/cu/enriquez/docencia/Quimica/vi.pdf>

- 8.1.13. Link=

<https://www.guru99.com/the-vi-editor.html>

- 8.1.15. link=

<https://linuxize.com/post/vim-find-replace/#:~:text=quick%20and%20easy.-,Basic%20Find%20and%20Replace,press%20the%20%27Esc%27%20key>

9.1.link=

<https://www.digitalocean.com/community/tutorials/how-to-read-and-set-environmental-and-shell-variables-on-linux-es#:~:text=Las%20variables%20de%20entorno%20son,se%20producen%20desde%20el%20shell.>

- 9.1.1. link 1=

<https://www.youtube.com/watch?v=E3btF4iUGpA>

- 9.1.1. link 2=

<https://computernewage.com/2015/06/14/el-arbol-de-directorios-de-linux-al-detalle-que-contiene-cada-carpeta/#:~:text=deber%C3%ADa%20contener%20binarios.,Home,las%20configuraciones%20de%20programas%2C%20etc>

- 9.1.1. link 3=

<http://www.w3big.com/es/linux/linux-comm-logname.html#:~:text=Linux%20LOGNAME%20comando%20muestra%20el.nombre%20del%20usuario%20actual.>

- 9.1.1. link 4=

<https://rootsudo.wordpress.com/2014/04/06/el-path-la-ruta-de-linux-variables-de-entorno/#:~:text=El%20PATH%20informa%20al%20shell,llamarlos%20por%20su%20ruta%20absoluta.>

&text=vemos%20que%20separados%20por%20dos,binarios%20(en%20ese%20orden).

- 9.1.1. link 5=

<https://www.solvetic.com/tutoriales/article/5109-como-usar-comando-hostname-en-linux/#:~:text=El%20comando%20hostname%20se%20implementa.es%20uno%20de%20los%20comandos.>

- 9.1.1. link 6=

https://www.linuxteaching.com/article/how_to_use_ifs_in_bash#what_is_ifs_in_bash
