|  |
| --- |
| **ARQUITECTURA Y SISTEMAS OPERATIVOS** |

|  |
| --- |
| **PROFESOR:** **JAVIER VASTAG Y ROBERTO MAXIMILIANO WENNER** |

|  |
| --- |
| **EN EL TURNO: MAÑANA** |

|  |
| --- |
| **TRABAJO PRÁCTICO N°1** |

|  |
| --- |
| **TÍTULO: LINUX** |

|  |  |
| --- | --- |
| **INTEGRANTES PRESENTES EL DÍA QUE SE REALIZÓ** | |
| MAXIMILIANO VILLANUEVA | DANIEL CASTRO |

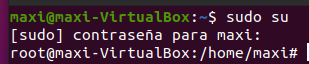
**Punto °1**

**Introducción**

* 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.

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



**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.

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.

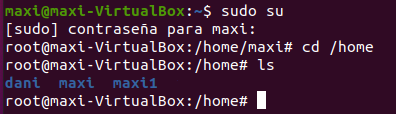
1.2. Ingresar al sistema como superusuario (root), y realizar los siguientes pasos:

1.5. **logout**

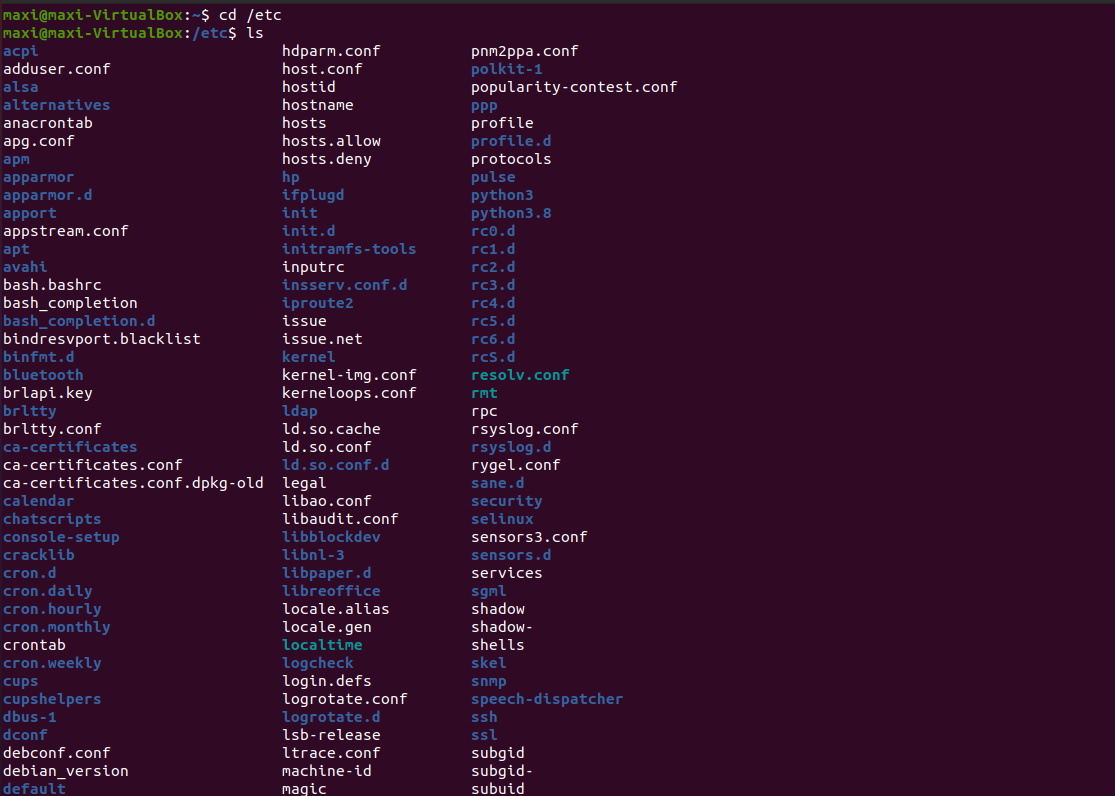
• **logout** = se utiliza para cerrar la terminal, pero en linux ubuntu se reemplaza por el comando **exit**, ya que no se utiliza en linux el comando Logout, y **logout** también se utiliza para salir de una cuenta inicia dentro de la terminal

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:



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





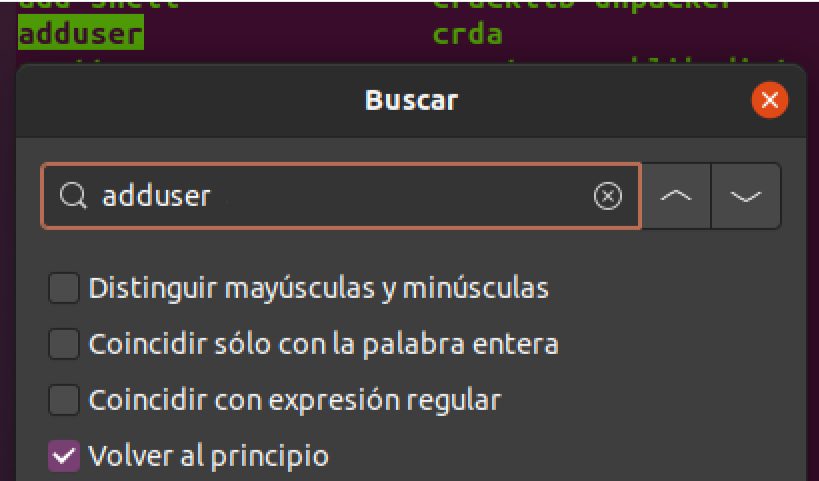
y para modificar la carpeta se utiliza el siguiente comando:



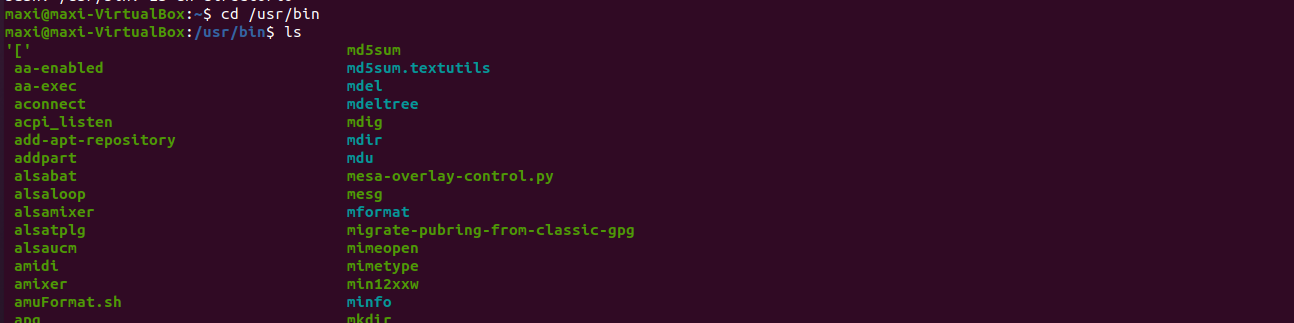
1.7. ¿En qué **directorio** se encuentran los comandos utilizados en los puntos 1.3 y 1.4?

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





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





**Punto °2**

**Teclado/Terminales**

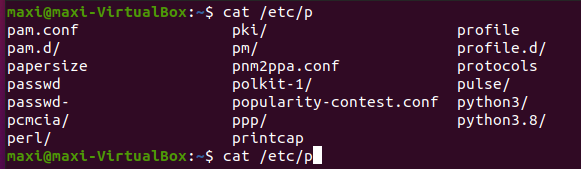
**2.TECLADO / TERMINALES**

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

cat /e <tab> p <tab>



Presione <tab> nuevamente ¿Qué pasó ahora?

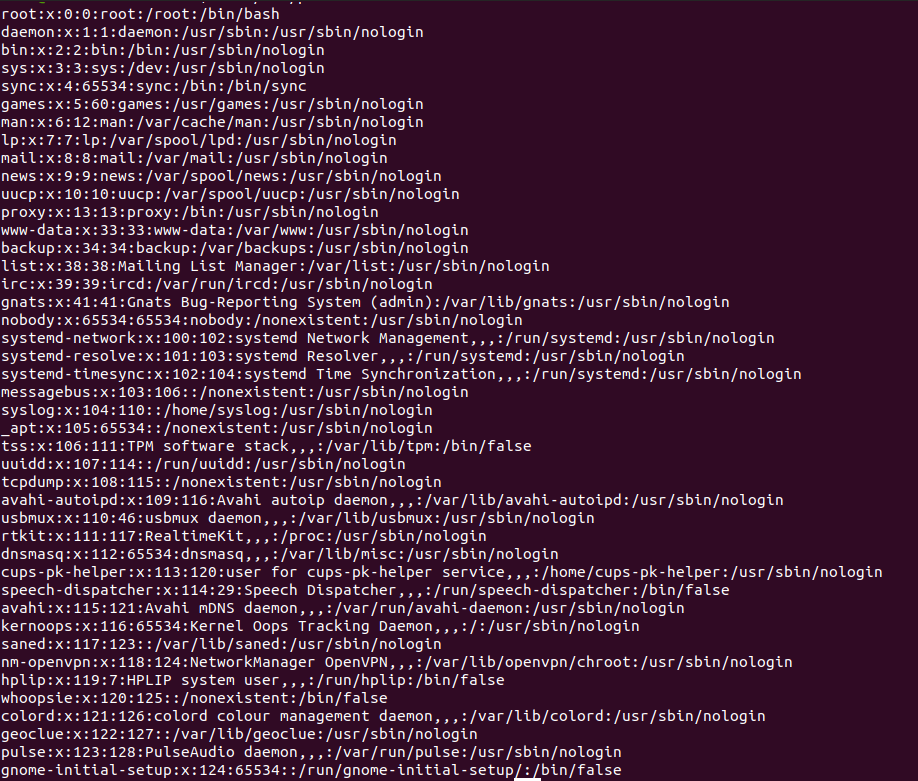


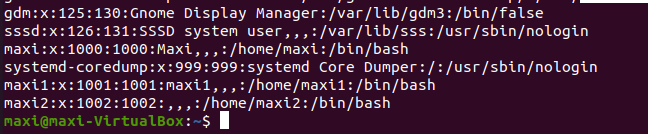
2.2. ¿Qué sucede si tecleo **cat /e <tab> pas <tab>?**

si lo escribo haciendo cada uno de sus pasos, pasa lo siguente:



Y me muestra esto:

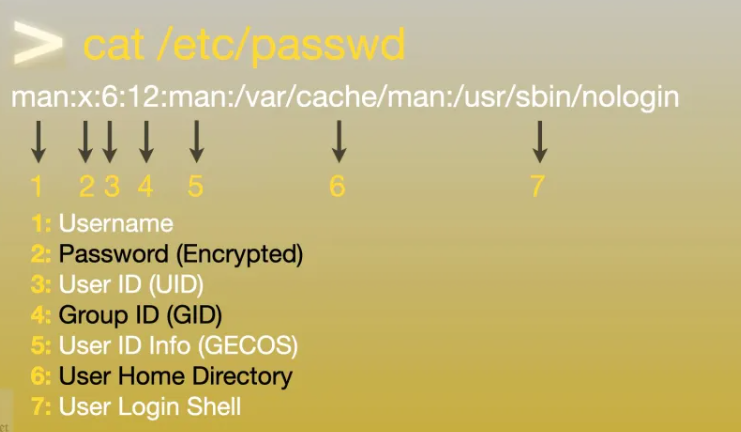




Explicación de lo sucedido:

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:**

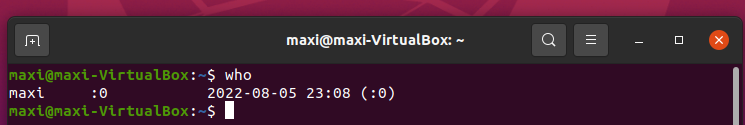
****

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.

1. **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.

**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?



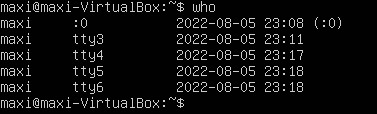
* La diferencia es que antes no aparecia **tty3** y ahora se muestra como un nuevo usuario.

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



* 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.



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

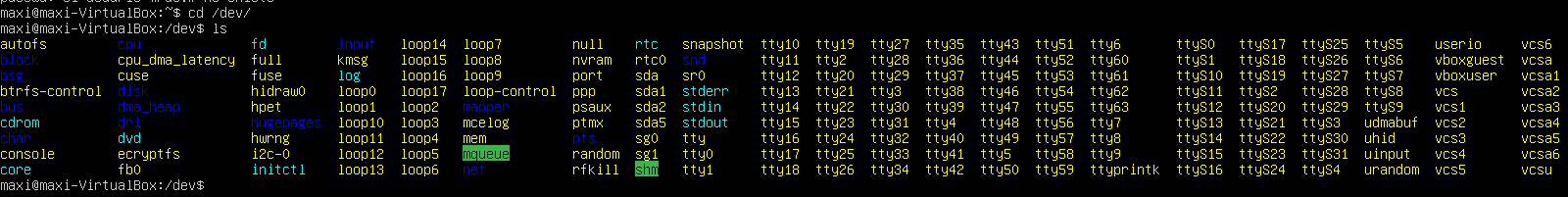
La **tty** es la consola que nos permite en GNU/Linux acceder a nuestro sistema operativo fuera de su entorno gráfico (X-Window). En general disponemos de hasta seis terminales.

Para acceder:

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

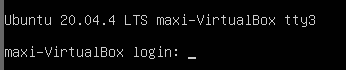
Para regresar al entorno gráfico: **Ctrl + Alt + F2**

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).





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**.

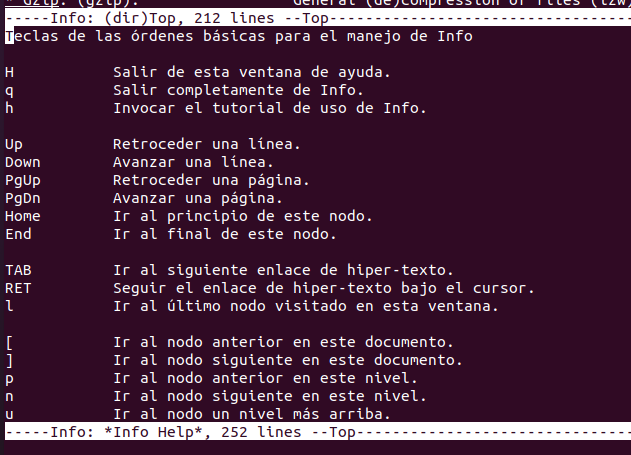
**Punto °3**

**Ayuda**

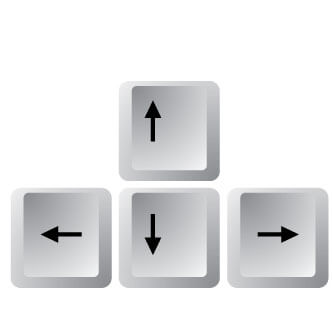
3.1. **INFO**: Info es un programa para leer documentación. Este se compone de una estructura del tipo árbol, dividido en nodos de información. Cada nodo describe un específico tópico con un determinado nivel de detalle.

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





y para ver los demás comandos puedes bajar usando las flechas, de la siguiente manera:



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?

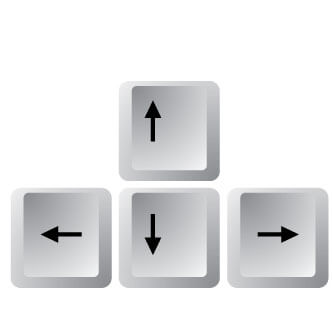
empezando desde **CP** vas bajando hasta encontrar próximo nodo que es **dd**



• **dd** : Convert and copy a file.

3.1.1.6. ¿Cómo puedo moverme al próximo nodo?

con teclas de direcciones, de la siguiente manera:



• se usa la tecla de dirección pulsando la tecla ⬇

También al presionar la tecla **N** te manda al próximo nodo.

3.1.1.7. ¿Cómo puedo moverme al nodo anterior?

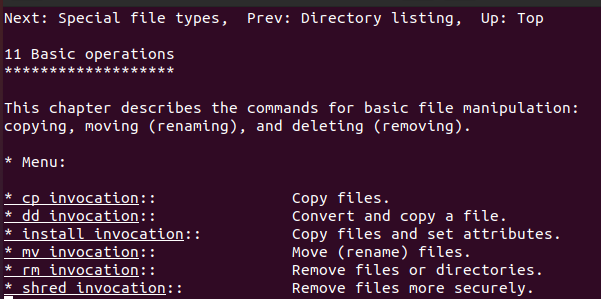
• con la tecla de dirección pulsando la tecla ⬆, y también presionando la tecla **L**

También al presionar la tecla **P** te manda al anterior nodo.

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”.

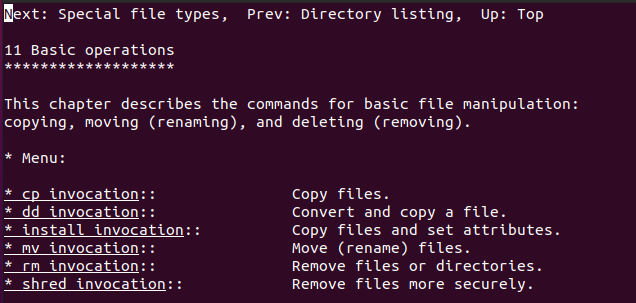


pero si no estás en ningún **nodo** te informa este mensaje:



3.1.2.2.  ¿En qué nodo se encuentra?

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



Pero antes nos encontrábamos en el nodo:



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 quieres buscar y apretas **ENTER**.

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

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



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

o también apretas la tecla **control**+**R** y escribes lo que quieres 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?



apretas la tecla **control**+**R** y escribes lo que quieres 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.

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

• **info [comando]**

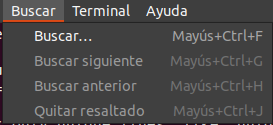
ejmplo: 

**Dato:**



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 **man**ual?





para poder buscar dentro del manual también se puede ingresar el comando “**ESC /**” y escribir lo que quieres 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?



• 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**

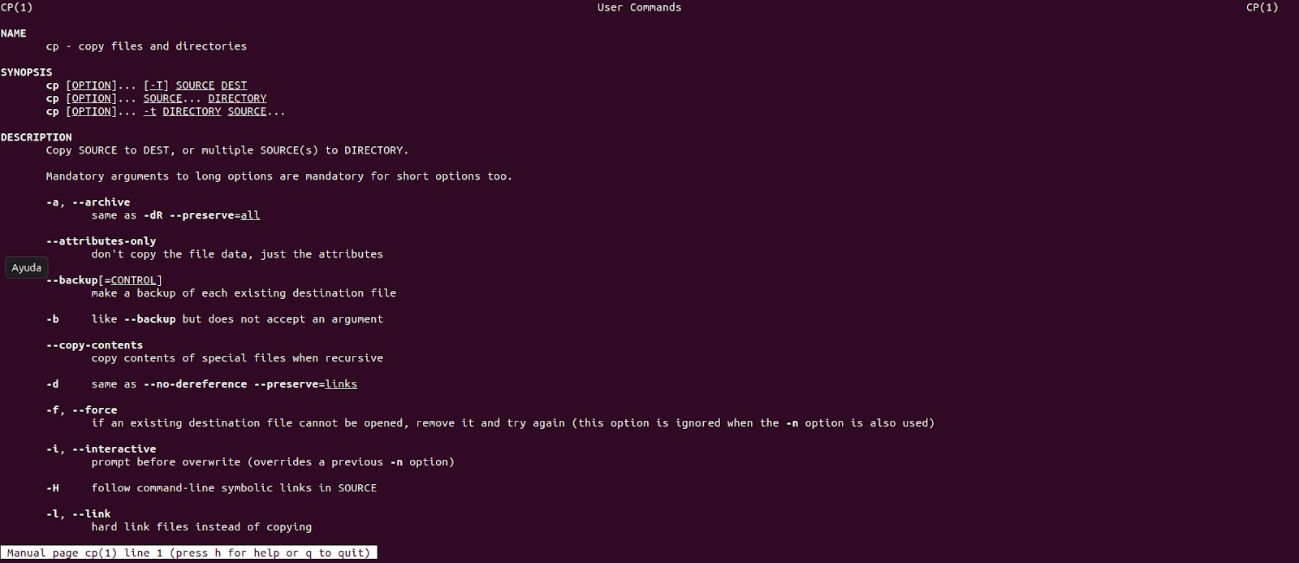


3.2.6.2. **man man**

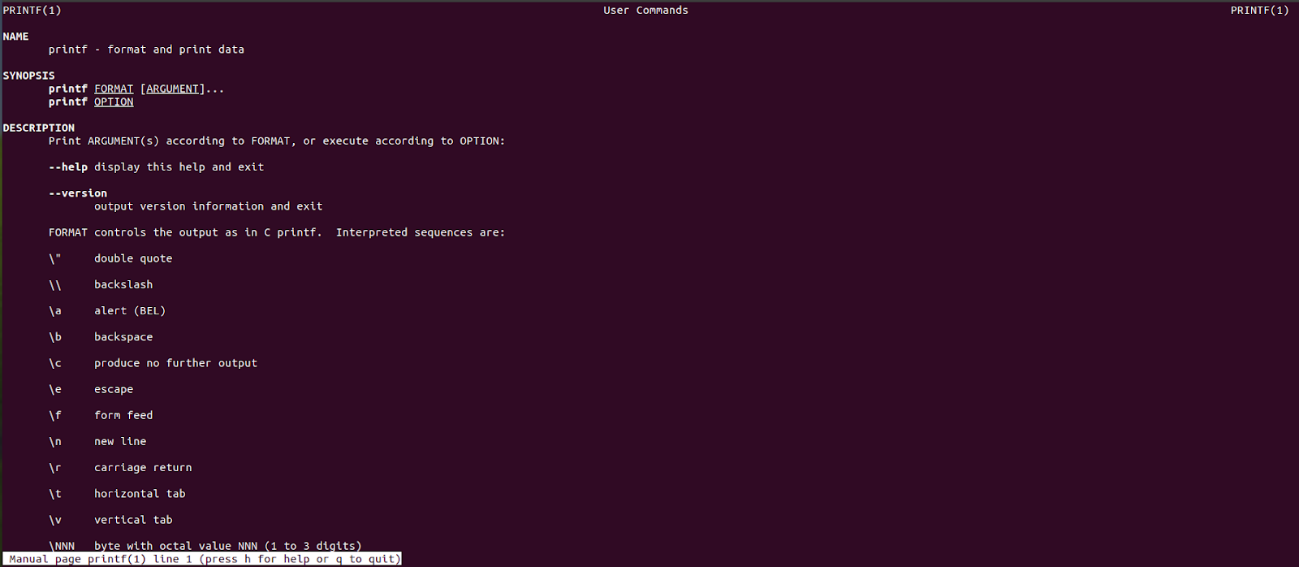
• abre la **interfaz** de los manuales de referencia del sistema de **man** y sus **SINOPSIS**

****

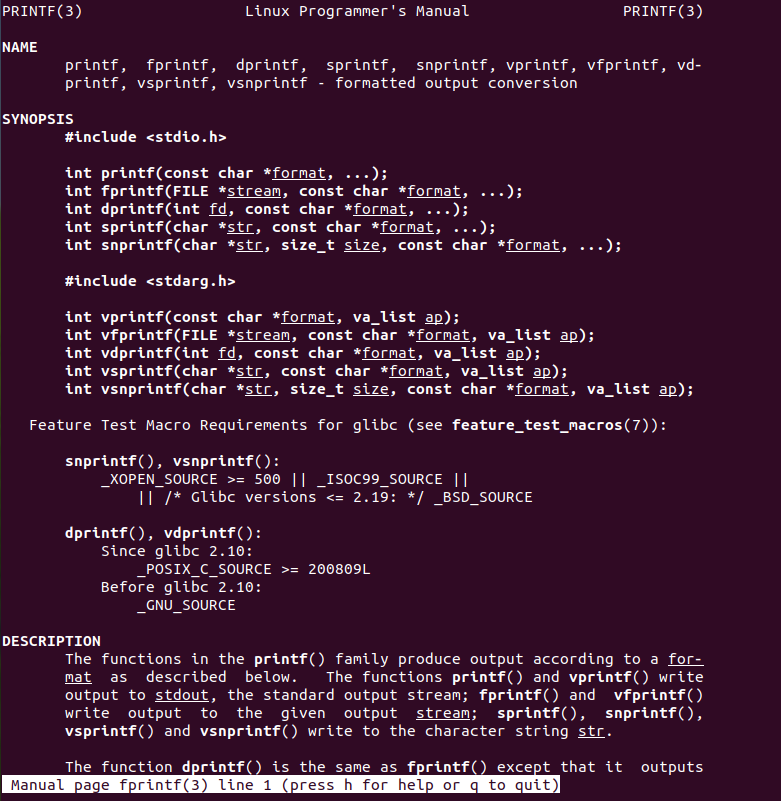
3.2.6.3. man cp



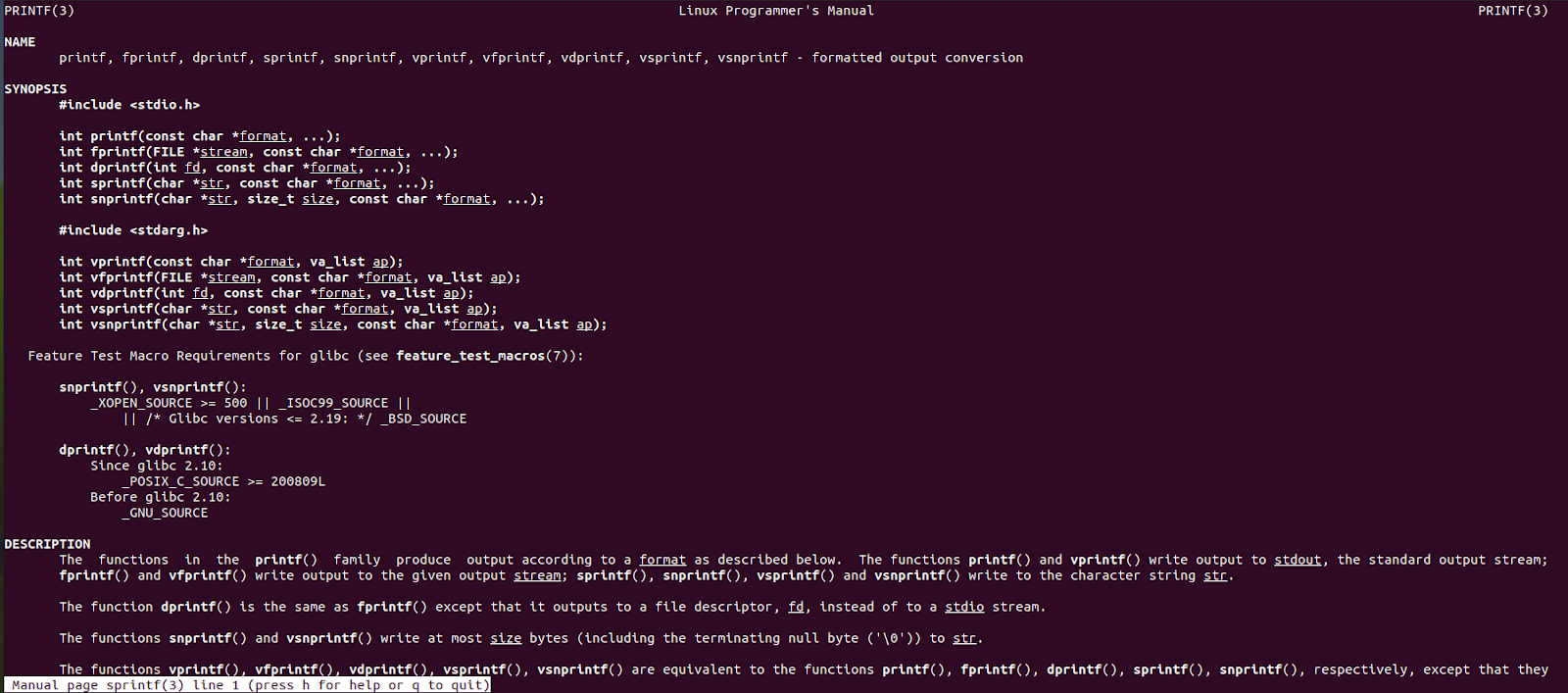
3.2.6.4. man printf



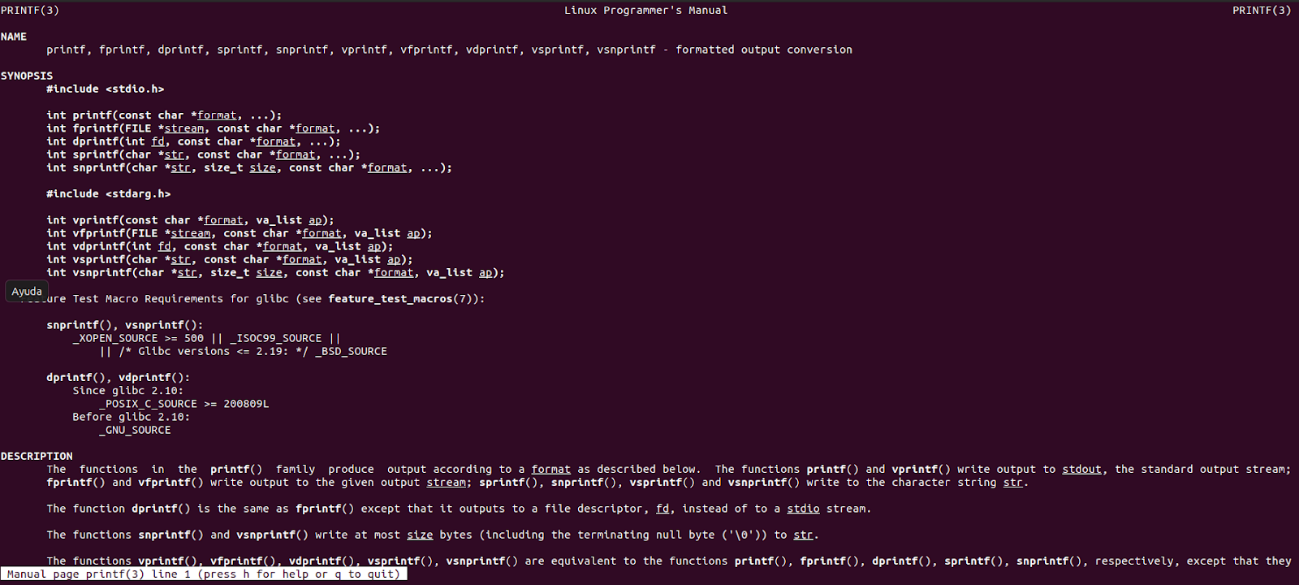
3.2.6.5. man fprintf



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).

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

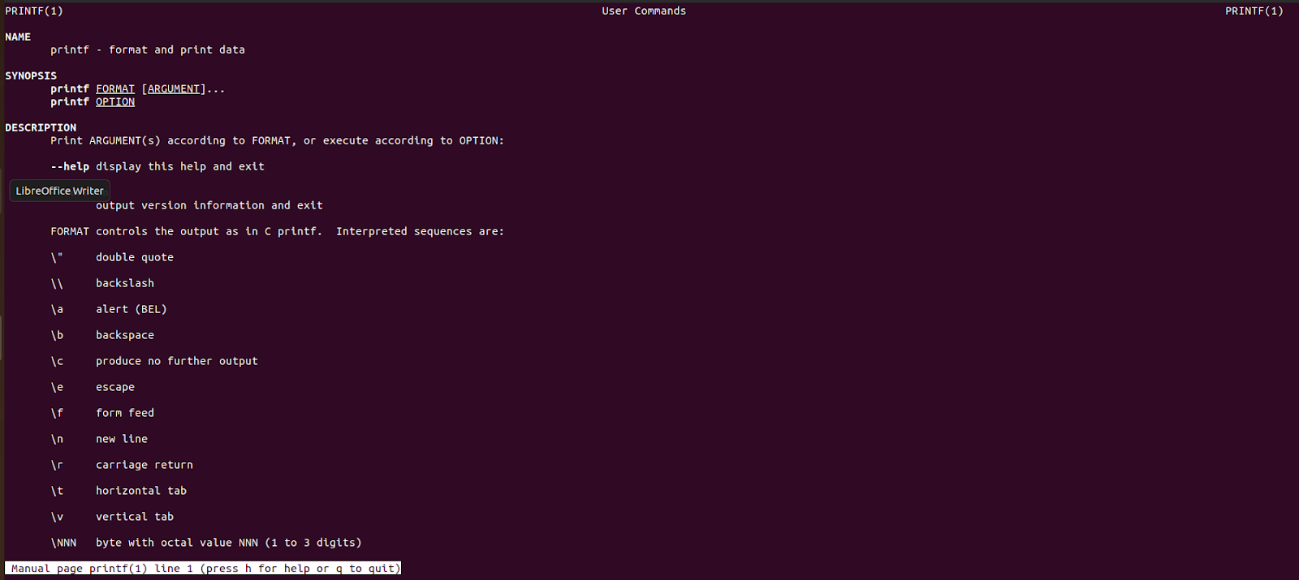
3.2.8. Del punto anterior, responder:

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”, correponden 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?.





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.

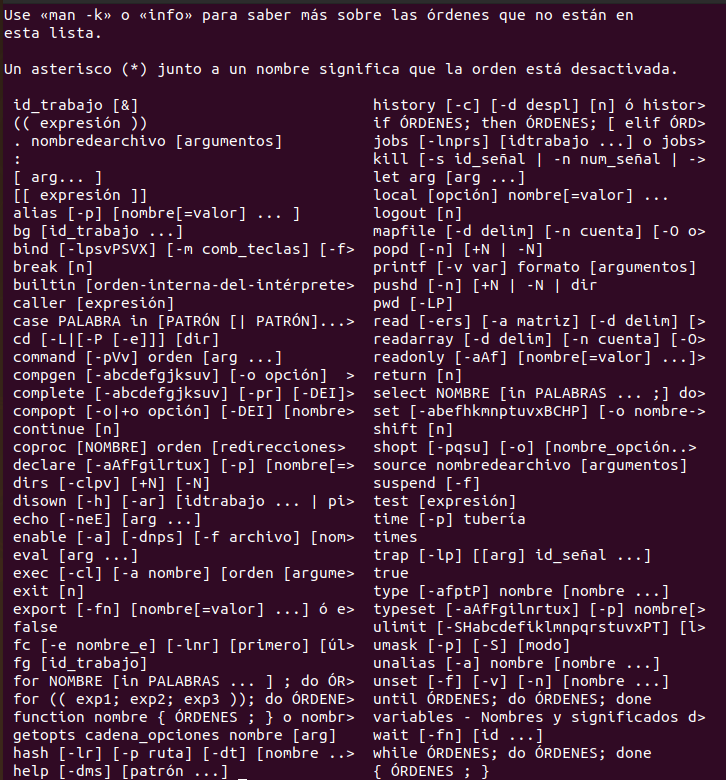
3.3.2. ¿Cuál es la diferencia entre **help** y **man**?.

**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.

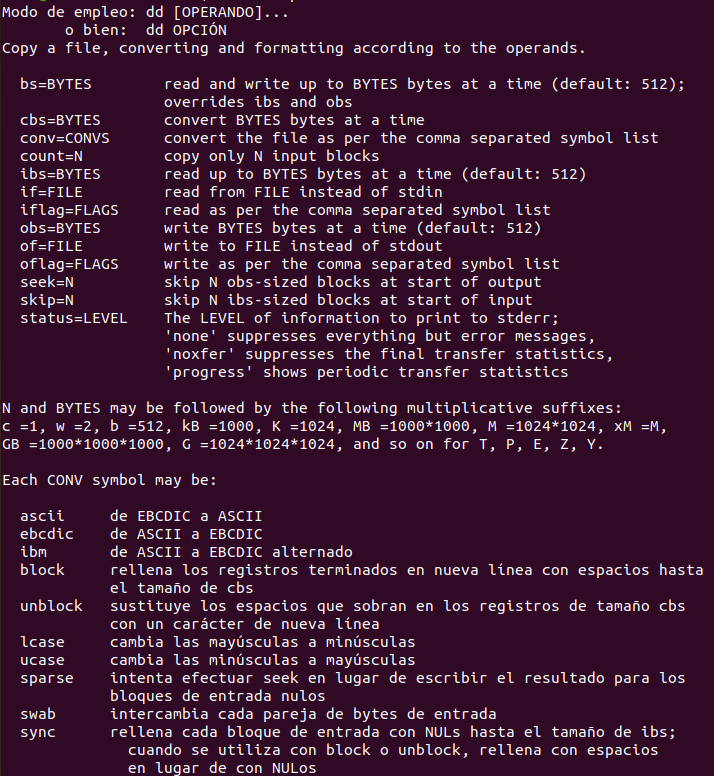
3.3.3. ¿Qué sucede al invocar al **help**?.



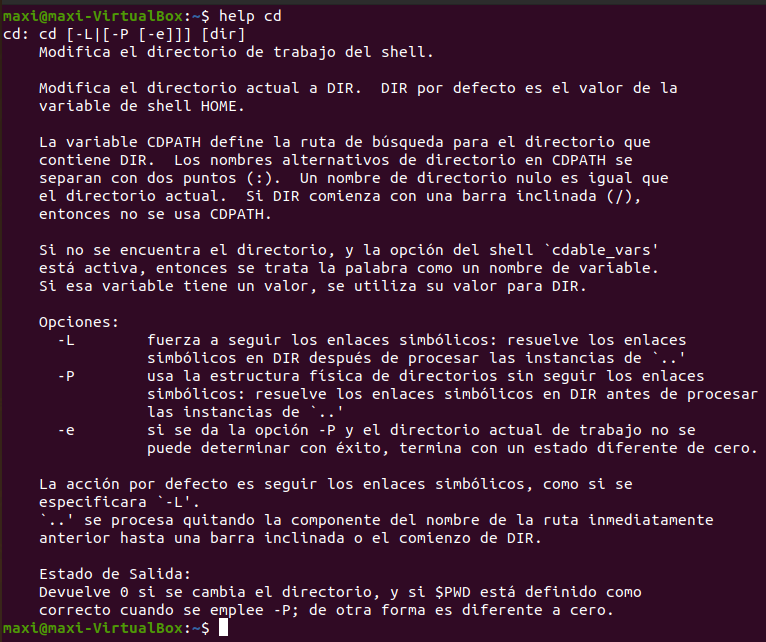
3.3.4. ¿Cómo puedo ver la información de un determinado comando?

• tienes que poner el comando y luego **–help,** ejemplo :





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**?

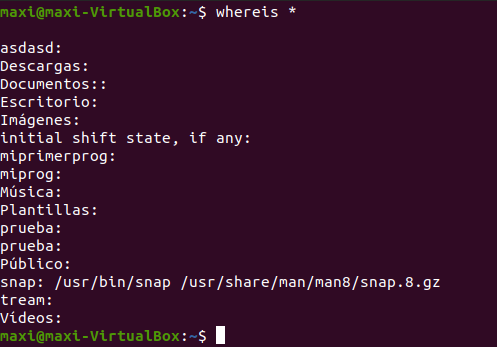


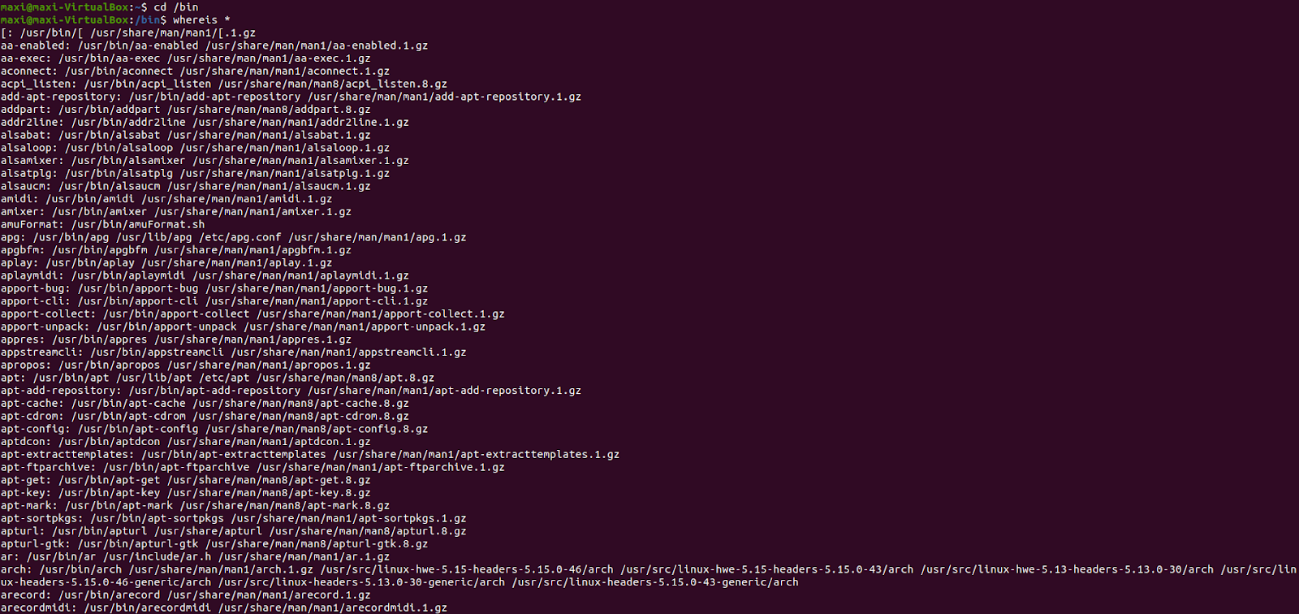
3.4.2. ¿Qué es la información que se muestra por pantalla al ejecutar el punto anterior?

• 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**)





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)

3.5. **whatis**

• **whatis:** lo que hace es explicar lo que hace un nodo

ejemplo:



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

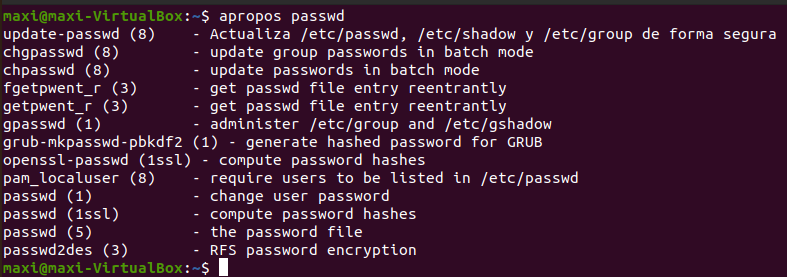


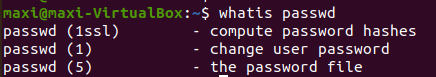
3.5.2. Cambie al directorio **/bin**, y ejecute el comando **whatis \*** ¿Qué ocurrió?





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





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

1. **apropos passwd** te da mayor información que **whatis passwd**

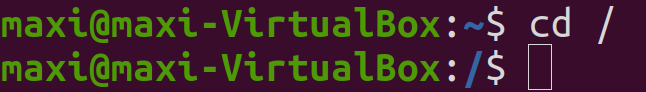
2. la información que ofrece **whatis passwd** está en **apropos passwd**

**Punto °4**

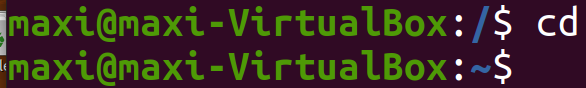
**Directorios**

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

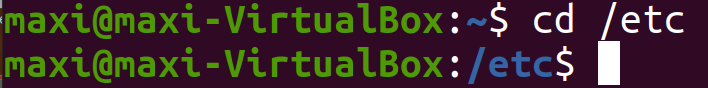
4.1.1. **cd /**



4.1.2. cd

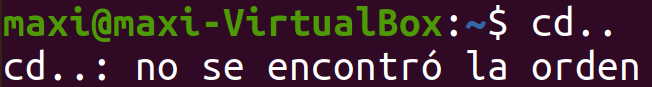


4.1.3. cd /etc

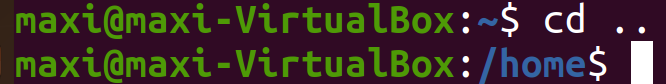


4.1.4. cd..

• se detecta como comando “cd..”.



4.1.5. cd ..



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

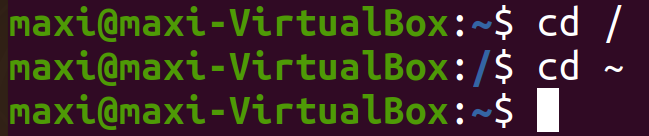
La ruta **absoluta** siempre comienza desde el directorio raíz (/). Por ejemplo, /home/indu/scripts/mis\_scripts.sh.

•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.

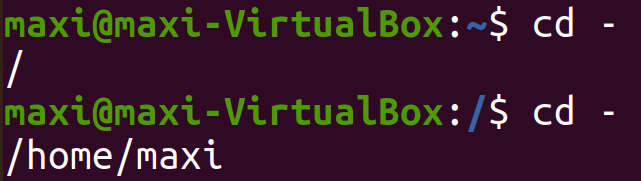
4.2.2. ¿Qué es lo que realizan las **siguientes** operaciones? (tip: si no encuentra la diferencia primero haga **cd /**, y luego vuelva a intentar)



4.2.2.1. cd ~



4.2.2.2. cd -



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

• 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?

• E comando para poder Crear Carpetas o Directorios es = **mkdir.**

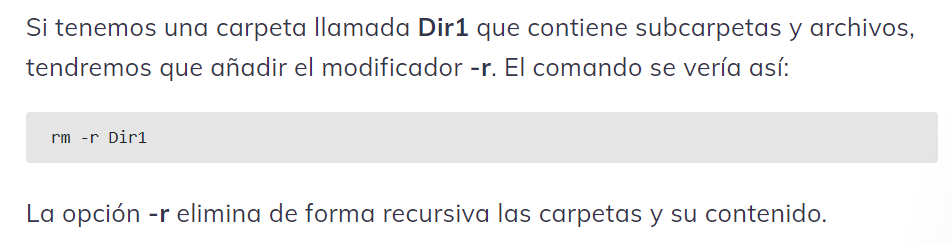
4.3.2. ¿Con qué comando se puede **borrar** un directorio?

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

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

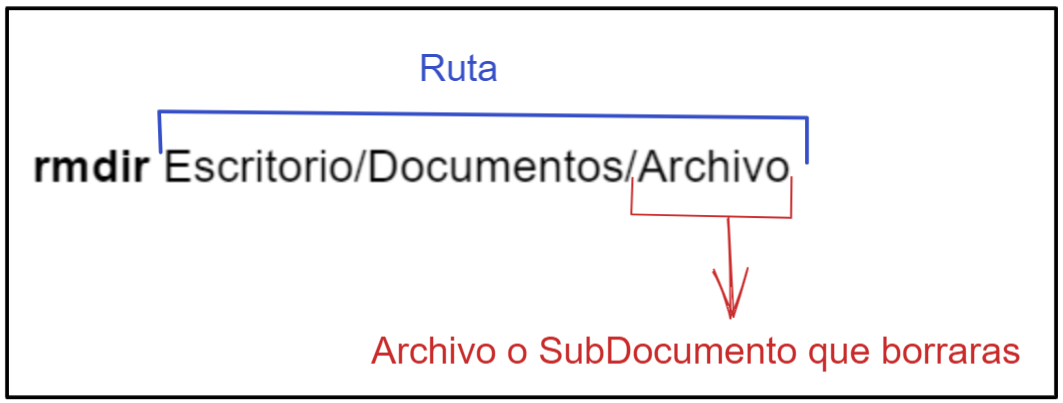
• puede haber subdirectorios y archivos.

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



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.

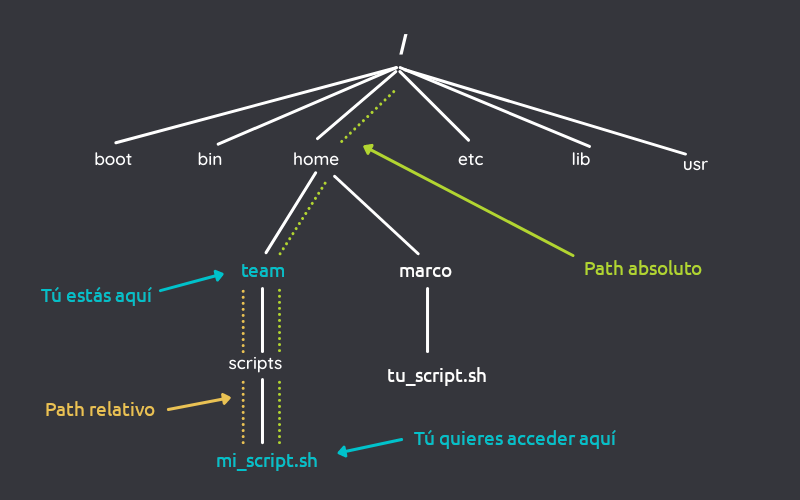
4.4. ¿Qué significa la expresión.**/** cuando se utiliza delante de un archivo?

• es para saber donde está ubicado los archivos.

¿Para qué sirve?

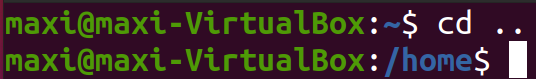
• sirve para saber el orden en que se abrieron las expresiones.

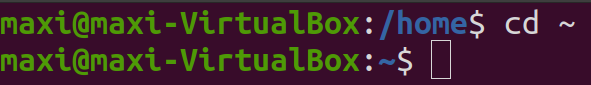
que es el **PATH** = “**/**”



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:





**Punto °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''.**

5.1.2. **mv**

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

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.

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** .

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.

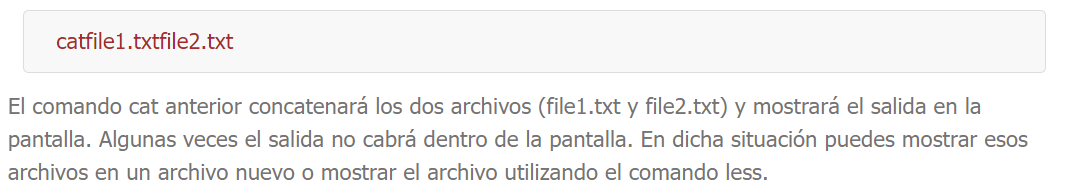
5.1.6. **scp**

• 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.

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.

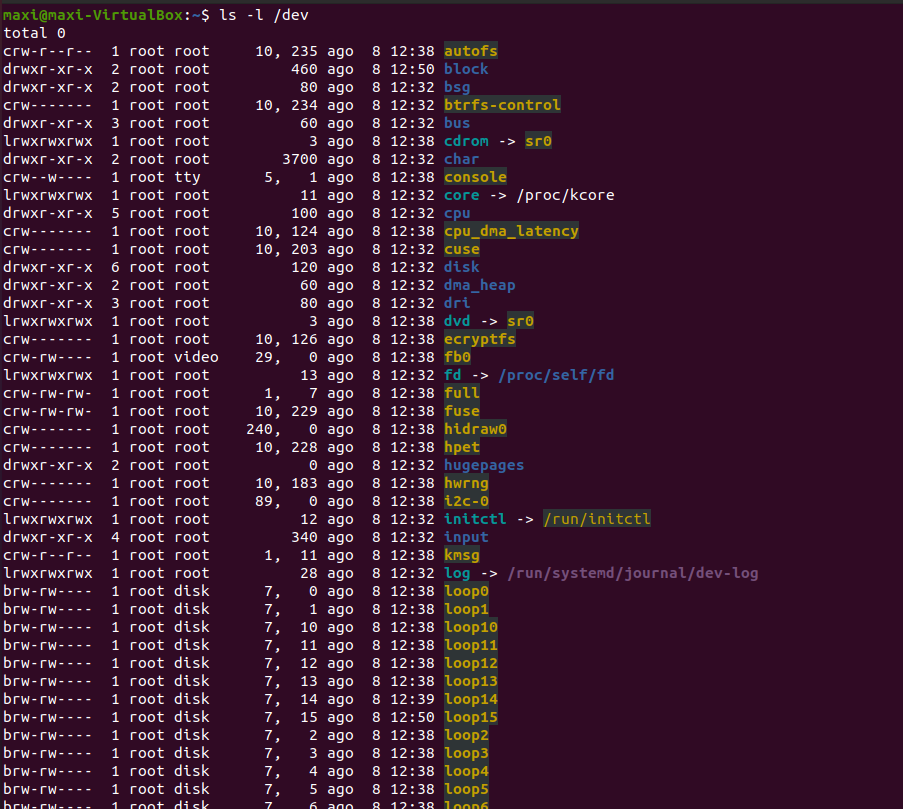
5.2. ¿Con qué comando puedo concatenar el contenido de dos archivos?



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

5.3.1. ¿Qué significa el primer carácter?

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

5.3.2. ¿Cuáles son todos los posibles valores que puede contener ese campo y que

¿Qué significa cada uno?

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

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

**Punto °6**

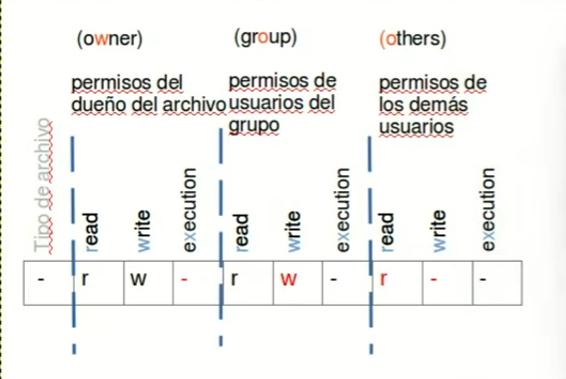
**Permisos**

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.

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 primer línea corresponde al propietario del archivo, la segunda línea corresponde a los usuarios del grupo y la tercera es para el resto de usuarios.

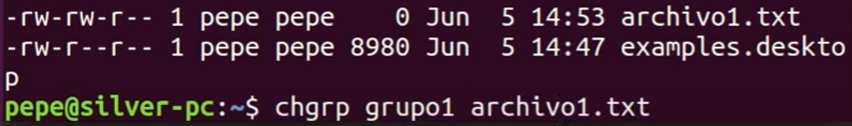
6.3. ¿Cómo se asignan los permisos? (detalle los comandos).



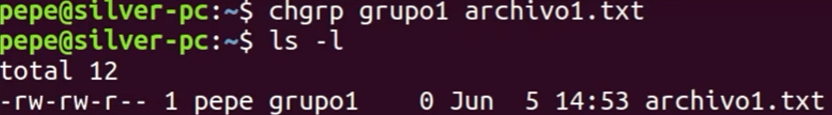
Primero se crean los grupos a los que les vamos a asignar esos permisos con el sudo goupadd mas el nombre que le queremos asignar..



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



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



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.

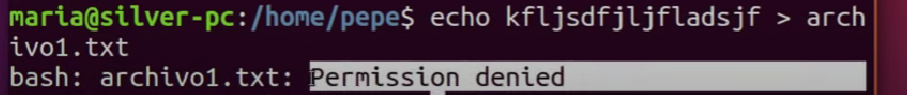
• **group**: Todos los usuarios que son miembros del mismo grupo.

¿Se pueden cambiar?

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

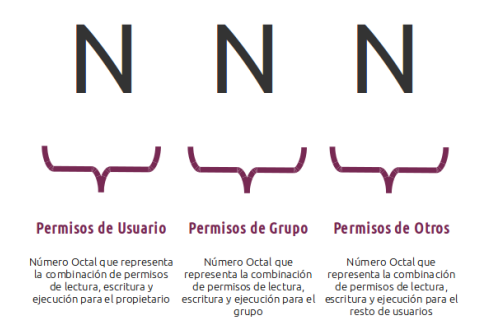
6.5. Intente cambiar los permisos de un archivo perteneciente al **root** (sesionado como **usuario**). Explique qué sucedió.

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

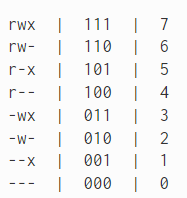


6.6. Explique la forma de cambiar los permisos con valores en octal.

• 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.



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).



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



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

**Punto °7**

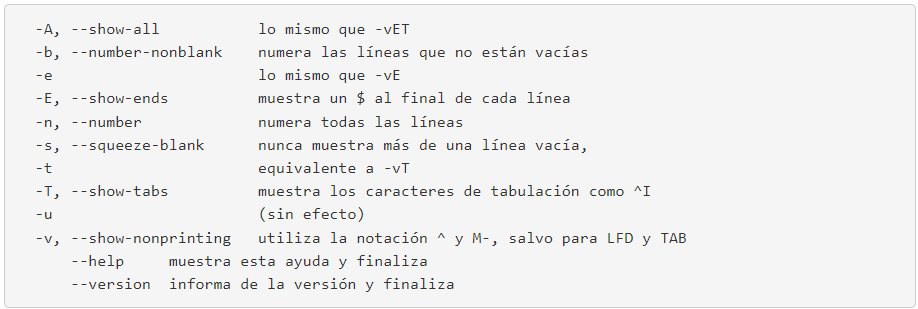
**Filtros**

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:



•**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.



•**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:



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.

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.

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

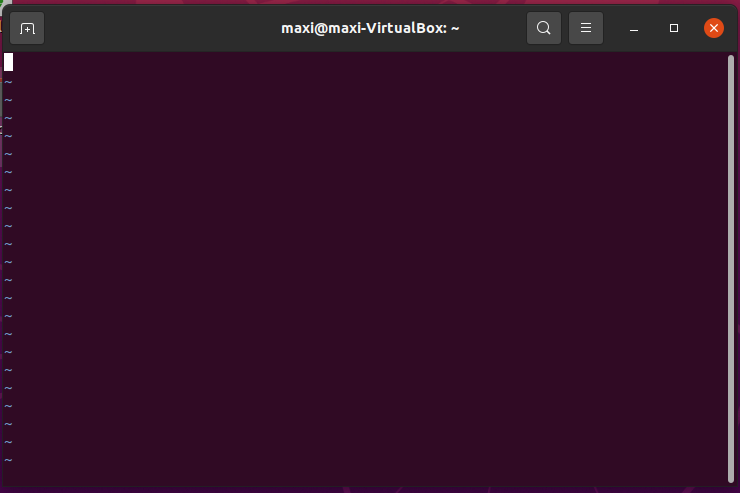
• **Grep** (por sus siglas en inglés Globally Search For Regular Expression and Print out) es una herramienta de líneas de comando usada para buscar cadenas de texto y encontrar coincidencias dentro de este. También puede ser utilizada para encontrar una palabra o combinación de palabras dentro de un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

**Punto °8**

**VI**

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



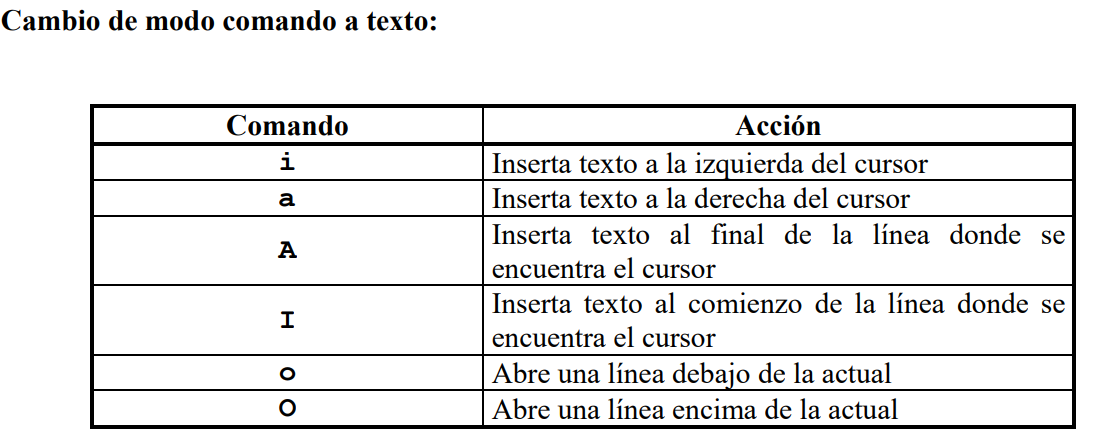


Lo que hace el comando **VI** es crear un archivo texto o tipo “**.txt**” que permite escribir dentro de la terminal.

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

• 

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



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?



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”.**



 y apretar la tecla **ENTER**.

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

**“**Sistemas Operativos

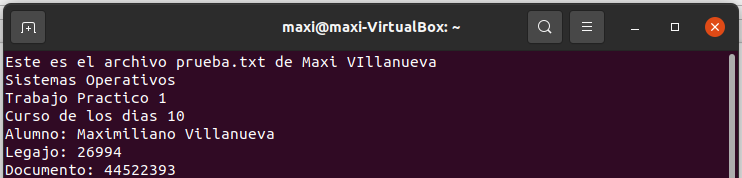
Curso de los días <día de cursada>

Trabajo Práctico 1

Alumno: <su nombre y apellido aquí>

Legajo: <su Legajo aquí>

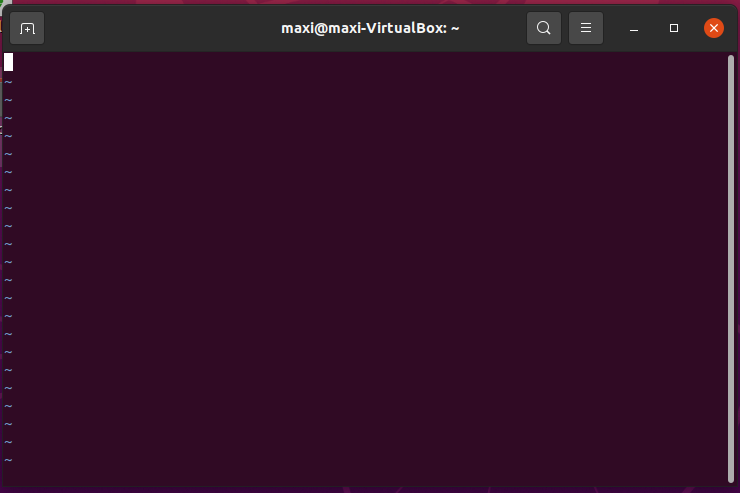
Documento: <su documento aquí>**”**



8.1.6. Describa todos los pasos que tuvo que realizar.

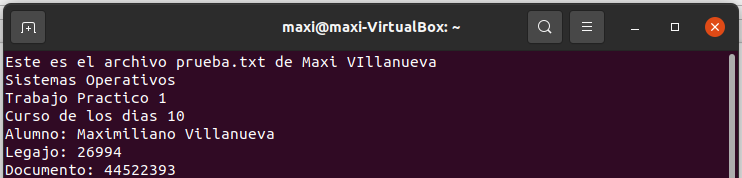
• Primer paso abrir la terminal y ejecutar el siguiente comando = “**vi $HOME/prueba.txt**”

• 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”.



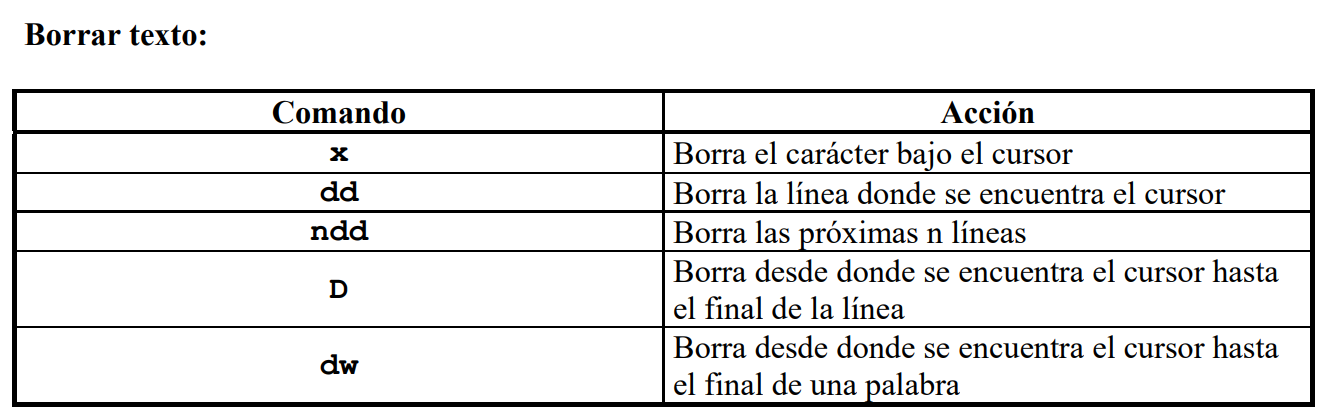
Una vez ingresado cada dato presionas la tecla “**Esc**” para luego ingresar el siguiente comando:



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

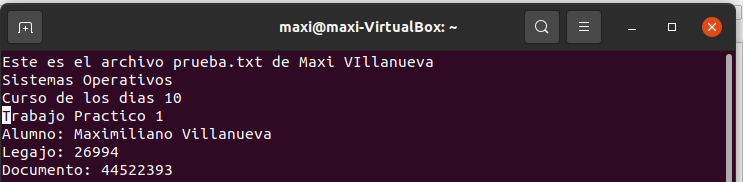


8.1.8. Borre la línea de “Legajo”. Indique por lo menos dos formas de realizarlo.



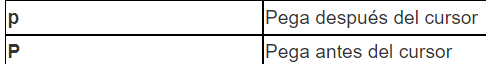
8.1.9. Invierta el orden de las líneas “**Curso** y **TP**”. No está permitido rescribirlas. ¿Qué comandos utilizó?

• estado actual del texto:



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

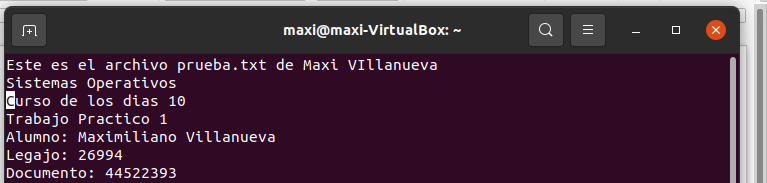




se puede usar “P” o “p”, sólo cambia la manera en que se pega el texto



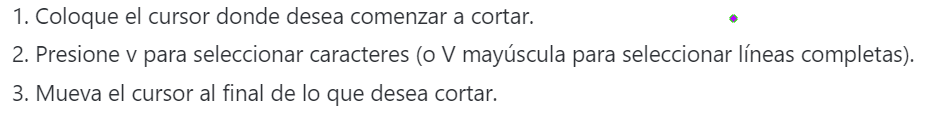
y quedaría el texto en la terminal así:



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 linera 2(dos) del archivo se debe de ingresar el siguente comando= **“:2”**

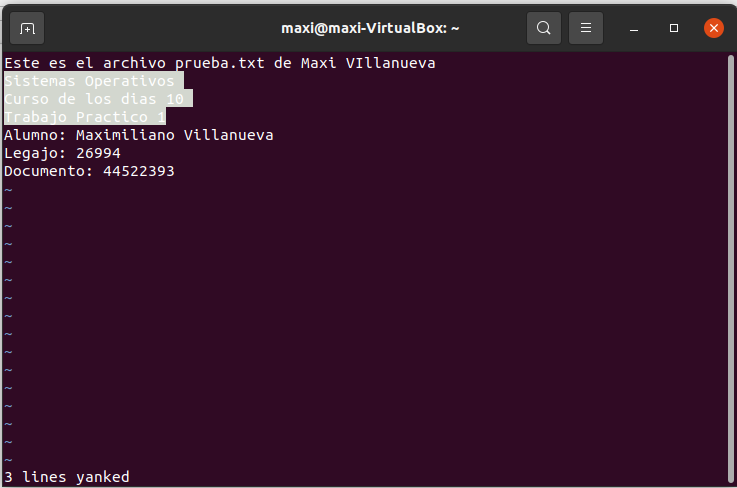
8.1.11. Marque para copiar las líneas 2, 3, y 4 (todas juntas, no de a una a la vez). ¿Cómo lo realizó?







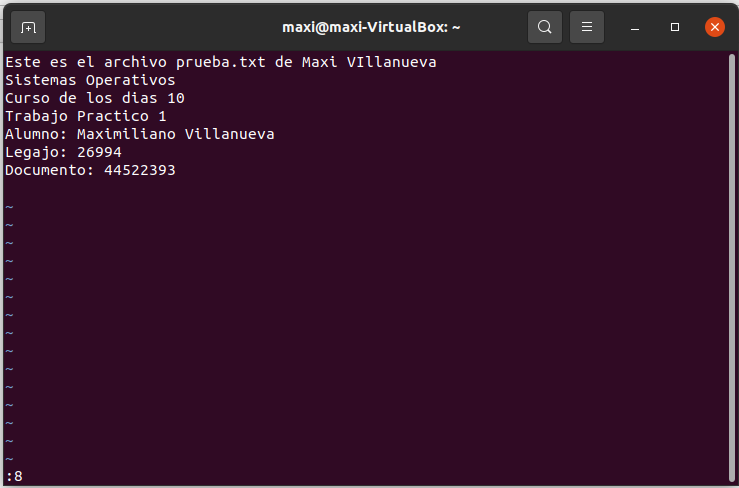




8.1.12. Ubíquese al final del archivo (sin usar las teclas del cursor), y pegue dos veces el contenido del **buffer**. ¿Qué comando usó?

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

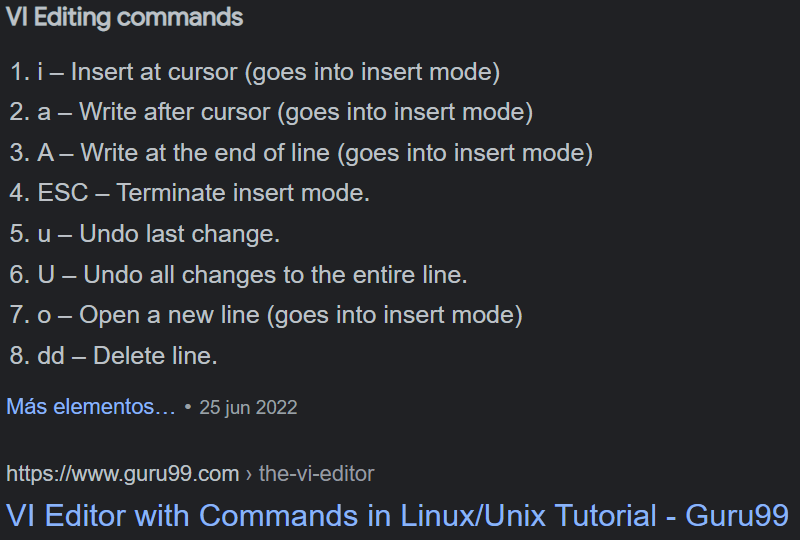
También para estar la última línea se puede utilizar el comando **Shift+g**.



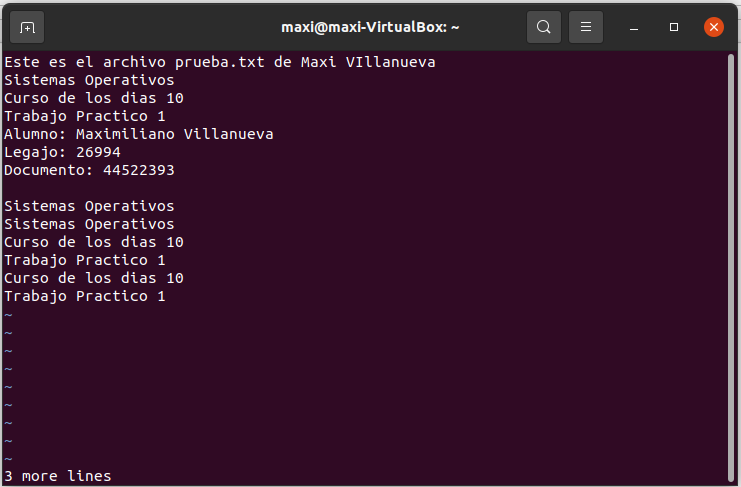


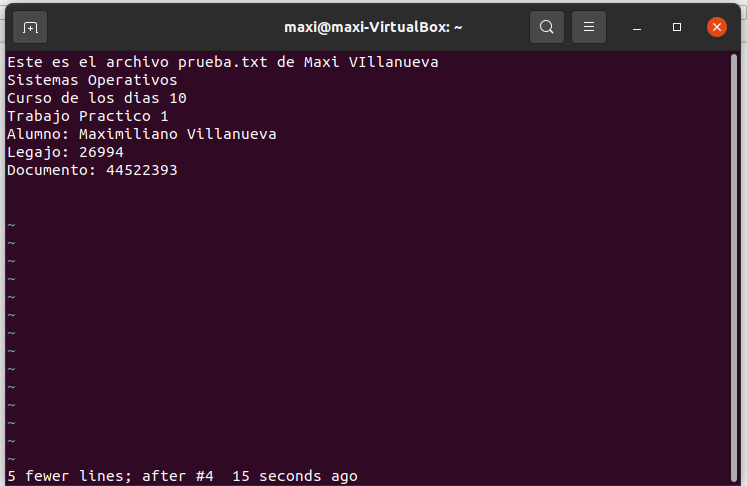


8.1.13. Deshaga uno de los copiados. No está permitido borrar línea por línea, ni carácter a carácter. ¿Qué comando usó?



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





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 ahi otra igual y si apretas “**n**”busca la palabra de **arriba** si ahi otra igual.

• ?Documento = busca desde la palabra de abajo, y luego apretar “**N**” busca la palabra de **arriba** si ahi otra igual y si apretas “**n**”busca la palabra de **abajo** si ahi 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

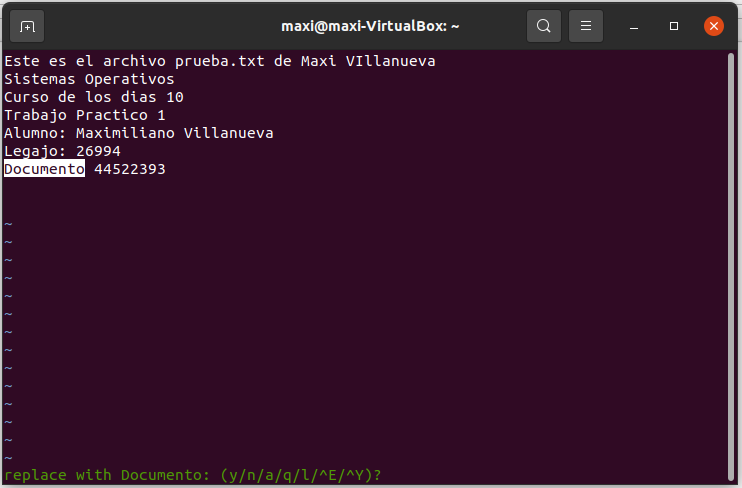
**g** = que busque todos los Documento y los remplace

**c** = que confirme todas las sustituciones y las reemplace

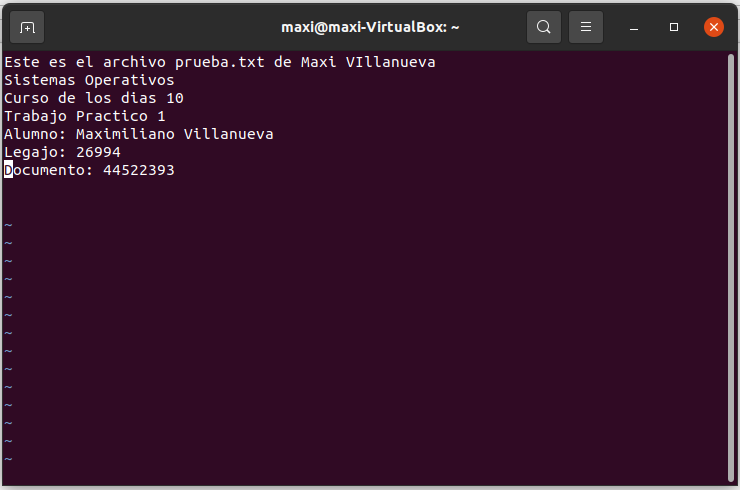


**y**=yes

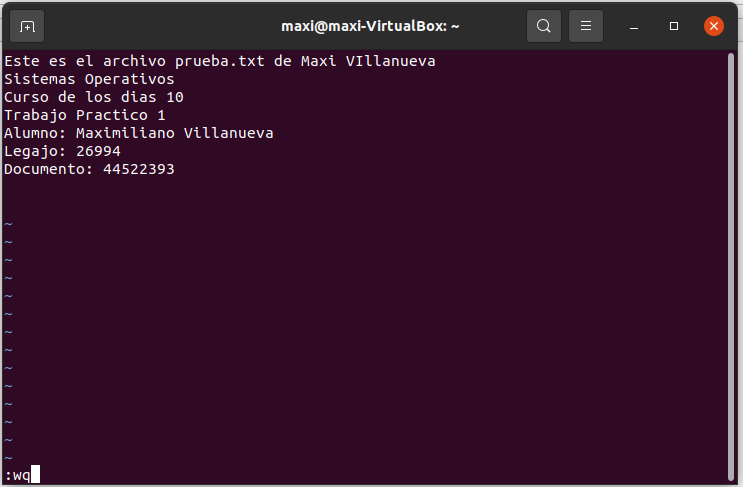
**n**=no



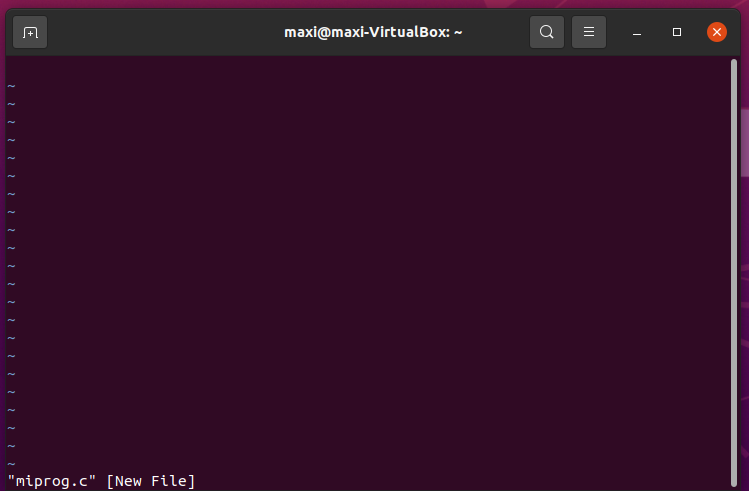
si quieres reemplazar la palabra poner **y**, si no quieres ingresar **n**



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");

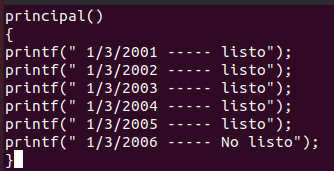
printf (" 1/3/2003 ------------- listo");

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**).

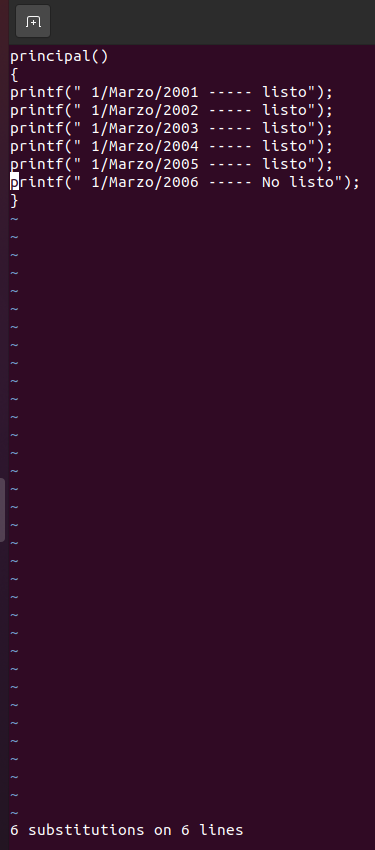


**./miprimerprog**

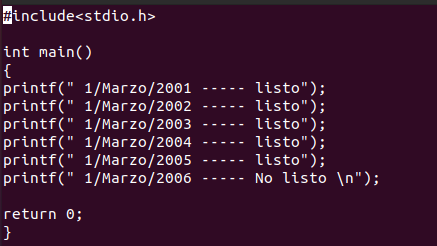
¿Cuál es la salida?

volver a ejecutar **vi miprog.c**

poner “**:%s/\/3/\/Marzo/g**” ¿Que pasó al ejecutar esto?



los errores que que corregir fueron los siguientes:



Corrija los errores que surgieron al utilizar el comando **gcc**.

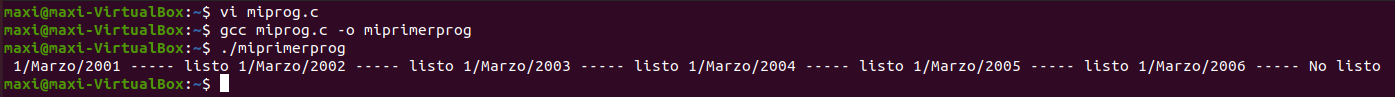
volver a ejecutar **:x**

ahora volver a ejecutar:

**gcc miprog.c -o miprimerprog**

**./miprimerprog**

¿Cual es la salida ahora?



**Punto °9**

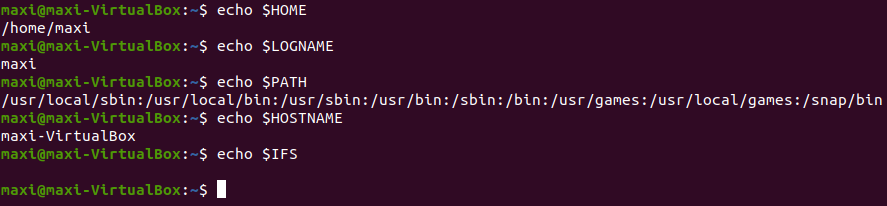
**Variables de entorno**

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.

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

• el contenido de las variables =



• 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.

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:



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



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.

**Conclusión**

Conclusión:

Esta guía fue un acercamiento del sistema operativo Linux, tanto lo teórico como lo práctico, este campo nos resultó totalmente nuevo, ya que, trabajamos totalmente sobre una consola/terminal y aprendiendo las funcionalidades de los variados comandos que posee este sistema operativo, dentro de esta guía pudimos observar y aprender sobre distintas áreas al momento de codificar, a la vez, entendimos que cualquier acción puede ser ejecutada desde el mismo terminal/consola, sin necesidad de realizarlo en la interfaz gráfica.

**Objetivo**

**Objetivo**:

El objetivo de esta guía, fue aprender y comprender sobre un tema en específico que en este caso es el sistema operativo Linux, sin ningún conocimiento previo. Investigando por nuestra cuenta por distintas partes virtuales como páginas web, manuales y/o videos, para poder resolver cada, ya sea, pregunta, ejercitación o problema a realizar.

Bibliografía:

1.1 Androide, Y. [YoAndroide]. (2020, May 21). UBUNTU: ENTRAR/ACTIVAR MODO ROOT/SUDO/SUPER USUARIO Y SALIR/DESACTIVAR. Youtube. <https://www.youtube.com/watch?v=J6Ah7Bqv09U>

1.3 Santos, J. A. S. [UCZ78PZ3mLZ0T2bVZAyyEyEg]. (2020, April 15). Comando ADDUSER en LINUX. Youtube. <https://www.youtube.com/watch?v=RyKlAnSGdHM>

2.2 /etc/passwd file in Linux. (2021, September 29). Cyber Sophia; CyberSophia.net.<https://cybersophia.net/linux/etc-passwd-file-in-linux/>

2.3.6 Miguel, J. (n.d.). Terminal tty en GNU/Linux. Tecno y Soft. Retrieved August 13, 2022, from <https://tecnoysoft.com/es/terminal-tty-en-gnulinux/>

3.2.1 ¿Cuál es el uso del comando man e Info en Linux? (2021, October 14). Compuhoy.com. <https://www.compuhoy.com/cual-es-el-uso-del-comando-man-e-info-en-linux/>

3.3.1 Difference between help, info and man command. (n.d.). Unix & Linux Stack Exchange. Retrieved August 13, 2022, from <https://unix.stackexchange.com/questions/19451/difference-between-help-info-and-man-command>

3.3.2 Navas, M. Á. (2016, July 23). Ayuda con los comandos linux desde el terminal. Profesional Review; Miguel Ángel Navas. <https://www.profesionalreview.com/2016/07/23/ayuda-comandos-linux/>

3.4.4 Difference among "whereis " , "locate " and “find”command. (n.d.). Ask Ubuntu. Retrieved August 13, 2022, from <https://askubuntu.com/questions/832562/difference-among-whereis-locate-and-findcommand>

4.2.1 Rutas absolutas y relativas en Linux: ¿Cuál es la diferencia? (2021, September 6). Técnicos. La Barra. <https://tecnicos.epet1.edu.ar/2021/09/rutas-absolutas-y-relativas-en-linux-cual-es-la-diferencia/>

4.3.3 Gustavo, B. (2019, August 20). Cómo borrar archivos y carpetas en Linux. Tutoriales Hostinger. <https://www.hostinger.com.ar/tutoriales/borrar-archivos-carpetas-linux>

5.1.1 Copiar en Linux: con CP es muy sencillo. (n.d.). IONOS Digitalguide. Retrieved August 13, 2022, from <https://www.ionos.es/digitalguide/servidores/configuracion/comando-cp-de-linux/>

5.1.2Wikipedia contributors. (n.d.). mv. Wikipedia, The Free Encyclopedia. <https://es.wikipedia.org/w/index.php?title=Mv&oldid=121920681>

5.1.3Wikipedia contributors. (n.d.). rm. Wikipedia, The Free Encyclopedia. <https://es.wikipedia.org/w/index.php?title=Rm&oldid=142315981>

5.1.4No title. (n.d.). Oracle.com. Retrieved August 13, 2022, from <https://docs.oracle.com/cd/E19620-01/805-7644/6j76kloqn/index.html>

5.1.5 No title. (n.d.). Oracle.com. Retrieved August 13, 2022, from <https://docs.oracle.com/cd/E19620-01/805-7644/6j76kloqp/index.html>

5.1.6 Copia remota con el comando scp - Gestión de sistemas remotos en Oracle Solaris 11.1. (2012, January 1). Oracle.com. <https://docs.oracle.com/cd/E37929_01/html/E36611/remotehowtoaccess-55154.html>

5.1.7 Comandos basicos de ssh. (n.d.). Uji.Es. Retrieved August 13, 2022, from <https://www3.uji.es/~galdu/ssh_vs_rsh/x165.html>

5.2 Comando Linux cat - Ejemplo de comando Concatenar Archivos, Tutorial Cat, Unix. (n.d.). Hscripts.com. Retrieved August 13, 2022, from <https://www.hscripts.com/es/tutoriales/linux-commands/cat.html>

5.3.1 Navas, M. Á. (2017, January 28). Permisos básicos en Linux: Ubuntu/Debian con CHMOD. Profesional Review; Miguel Ángel Navas. <https://www.profesionalreview.com/2017/01/28/permisos-basicos-linux-ubuntu-chmod/>

5.3.2 (1) WhileTrueThenDream [WhileTrueThenDream]. (2020, June 5). Permisos, usuarios y grupos en Linux. (Administración básica). Youtube.

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

5.3.2 (2) Navas, M. Á. (2017, January 28). Permisos básicos en Linux: Ubuntu/Debian con CHMOD. Profesional Review; Miguel Ángel Navas. <https://www.profesionalreview.com/2017/01/28/permisos-basicos-linux-ubuntu-chmod/>

5.4 Alex. (n.d.). touch - crear o actualizar ficheros. Cambiatealinux.com. Retrieved August 13, 2022, from <https://cambiatealinux.com/touch-crear-o-actualizar-ficheros>

6.1 Navas, M. Á. (2017, January 28). Permisos básicos en Linux: Ubuntu/Debian con CHMOD. Profesional Review; Miguel Ángel Navas. <https://www.profesionalreview.com/2017/01/28/permisos-basicos-linux-ubuntu-chmod/>

6.2 Navas, M. Á. (2017, January 28). Permisos básicos en Linux: Ubuntu/Debian con CHMOD. Profesional Review; Miguel Ángel Navas. <https://www.profesionalreview.com/2017/01/28/permisos-basicos-linux-ubuntu-chmod/>

6.3 Introducción a la asignación de permisos de directorio en Linux. (n.d.). IONOS Digitalguide. Retrieved August 13, 2022, from <https://www.ionos.es/digitalguide/servidores/know-how/asignacion-de-permisos-de-acceso-con-chmod/>

6.4 Gustavo, B. (2018, May 30). Cómo cambiar permisos y propietarios en Linux a través de la línea de comandos. Tutoriales Hostinger. <https://www.hostinger.com.ar/tutoriales/cambiar-permisos-y-propietarios-linux-linea-de-comandos/>

6.5 WhileTrueThenDream [WhileTrueThenDream]. (2020, June 5). Permisos, usuarios y grupos en Linux. (Administración básica). Youtube. <https://www.youtube.com/watch?v=KhbM0BQ3_ew>

6.6 Oriol, V. T. las E. (2015, June 27). Conoce la Arquitectura de Permisos de Linux al Detalle. ComputerNewAge. <https://computernewage.com/2015/06/27/conoce-la-estructura-de-permisos-de-linux-al-detalle/>

6.7 Oriol, V. T. las E. (2015, June 27). Conoce la Arquitectura de Permisos de Linux al Detalle. ComputerNewAge. <https://computernewage.com/2015/06/27/conoce-la-estructura-de-permisos-de-linux-al-detalle/>

7.1 Linux: diferencias entre los comandos cat, more y less. (2010, April 5). # rm-rf.es | Blog de un sysadmin Unix, GNU/Linux, Windows y lo que haga falta; # rm-rf.es. <https://rm-rf.es/unix-diferencias-entre-los-comandos-cat-more-y-less/>

7.2(N.d.). Platzi.com. Retrieved August 13, 2022, from <https://platzi.com/clases/1667-linux/22993-diferencias-entre-less-cat-head-y-tail-paralectur/#:~:text=Tail%3A%20muestra%20la%20ultima%20parte,parametros%20de%20lineas%20a%20ver>

7.3 (N.d.). Compuhoy.com. Retrieved August 13, 2022, from <https://www.compuhoy.com/como-se-usa-uniq-linux/#%C2%BFComo_usar_el_comando_Uniq_en_Linux>

7.4 ¿Qué es el comando Grep y cómo usarlo en Linux? (2022, February 28). KeepCoding Tech School. <https://keepcoding.io/blog/que-es-el-comando-grep-y-como-usarlo-en-linux/>

8. (N.d.). Unirioja.Es. Retrieved August 13, 2022, from <https://www.unirioja.es/cu/enriquez/docencia/Quimica/vi.pdf>

8.1.9 vi - referencia rápida. (n.d.). Com.mx. Retrieved August 14, 2022, from <https://www.linuxtotal.com.mx/index.php?cont=info_admon_010>

8.1.11(1) ¿Cómo se copian varias líneas en Linux? (2021, September 18). Compuhoy.com. <https://www.compuhoy.com/como-se-copian-varias-lineas-en-linux/>

8.1.11(2) vi - referencia rápida. (n.d.). Com.mx. Retrieved August 14, 2022, from <https://www.linuxtotal.com.mx/index.php?cont=info_admon_010>

8.1.11(3) (N.d.). Unirioja.Es. Retrieved August 14, 2022, from <https://www.unirioja.es/cu/enriquez/docencia/Quimica/vi.pdf>

8.1.12 (N.d.). Unirioja.Es. Retrieved August 14, 2022, from <https://www.unirioja.es/cu/enriquez/docencia/Quimica/vi.pdf>

8.1.13 Brent, M. (2020, March 11). VI editor with commands in Linux/Unix tutorial. Guru99. <https://www.guru99.com/the-vi-editor.html>

8.1.15 (N.d.). Linuxize.com. Retrieved August 14, 2022, from <https://linuxize.com/post/vim-find-replace/#:~:text=quick%20and%20easy.-,Basic%20Find%20and%20Replace,press%20the%20%27Esc%27%20key>

9.1 Ellingwood, J. (2020, December 2). Cómo leer y configurar variables de entorno y de shell en Linux. Digitalocean.com; DigitalOcean. <https://www.digitalocean.com/community/tutorials/how-to-read-and-set-environmental-and-shell-variables-on-linux-es>

9.1.1(1) Oriol, V. T. las E. (2015, June 14). El Árbol de Directorios de Linux. Conoce las Principales Carpetas. ComputerNewAge. <https://computernewage.com/2015/06/14/el-arbol-de-directorios-de-linux-al-detalle-que-contiene-cada-carpeta/>

9.1.1(2) comandos de Linux logname. (n.d.). W3big.com. Retrieved August 14, 2022, from <http://www.w3big.com/es/linux/linux-comm-logname.html>

9.1.1(3) (N.d.). Wordpress.com. Retrieved August 14, 2022, from <https://rootsudo.wordpress.com/2014/04/06/el-path-la-ruta-de-linux-variables-deentorno/#:~: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(4) Sistemas, S. (2018, March 21). Cómo usar comando Hostname en Linux. Solvetic. <https://www.solvetic.com/tutoriales/article/5109-como-usar-comando-hostname-en-linux/>

9.1.1(5) Lucio, V. (n.d.). Cómo usar $ IFS en Bash? Linuxteaching. Retrieved August 14, 2022, from <https://www.linuxteaching.com/article/how_to_use_ifs_in_bash>

9.1.3 Daza, Y. [YeisonDaza]. (2016, April 4). Variables de entorno en Linux. Youtube. <https://www.youtube.com/watch?v=VSmfKeTkL48>

9.1.5Comandos multilínea. (n.d.). Seagullscientific.com. Retrieved August 14, 2022, from<https://help.seagullscientific.com/2021/es/subsystems/btvbscript/content/sib_multi_line_scripts.htm>

Índice:

Punto1………………………………………………………………………….…..…1

Punto2………………………………………………………………………….…..…6

Punto3………………………………………………………………………….…..…13

Púnto4………………………………………………………………………….…..…30

Punto5………………………………………………………………………….…..…35

Punto6………………………………………………………………………….…..…39

Punto7………………………………………………………………………….…..…44

Punto8……………………………………………………………………………..….47

Punto9………………………………………………………………………….…..…62

Conclusiones………………………………………………………………………….…..…65

Objetivos………………………………………………………………………….…..…67

Bibliografia ……………………...……………………………………………………...69