
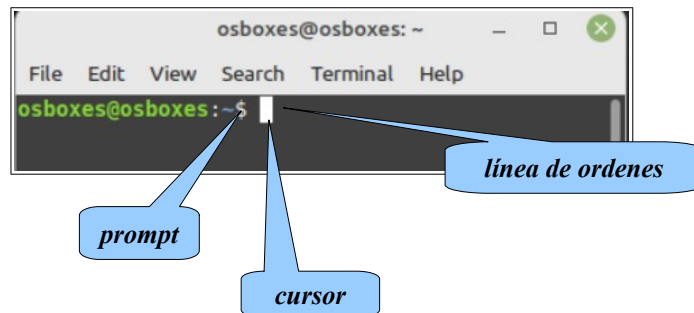


**PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ**  
**FACULTAD DE CIENCIAS E INGENIERÍA**  
**INGENIERÍA INFORMÁTICA**  
**SISTEMAS OPERATIVOS**

**Línea de Ordenes**

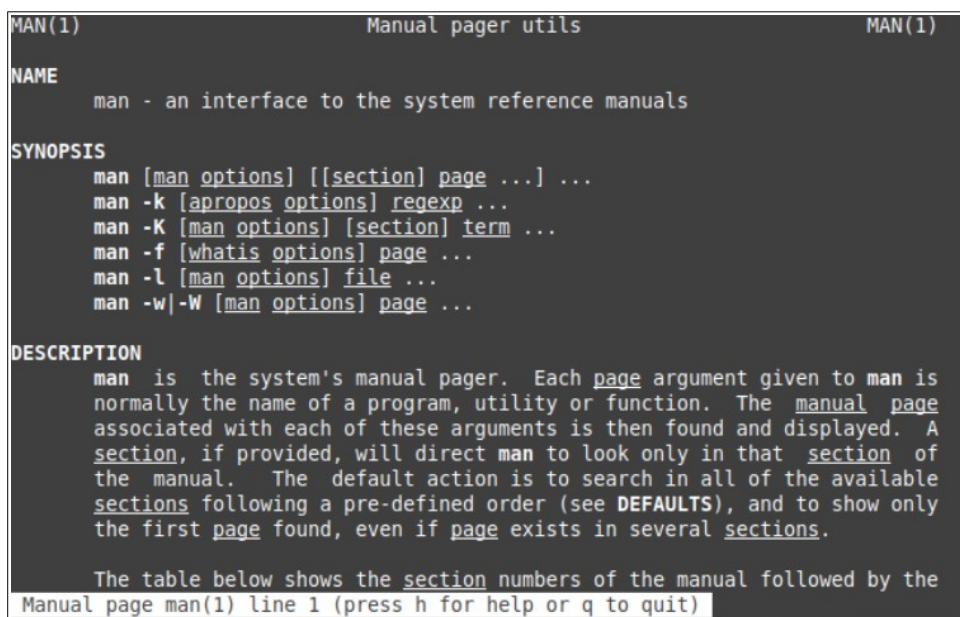
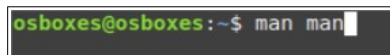
En este primer laboratorio todos los ejercicios se llevará a cabo en una terminal virtual (  ). Presione Ctrl + Alt + t para obtener la *Terminal*.



Para ejecutar cualquier orden, primero debe de escribirlo en la terminal y luego presionar la tecla <Enter>. Muchas ordenes permiten la especificación de opciones al momento de ejecutarlo. Estas opciones normalmente van precedidas por un guión (-) seguidas de una letra. Por ejemplo en la orden `ls -l`, la opción `-l` indica que debe mostrarse el listado en un formato largo.

Usted puede obtener una explicación más detallada con el manual de ayuda en línea: **man**. El manual en línea es de gran utilidad durante una sesión, si desea información acerca de alguna orden, puede invocarlo de la siguiente forma: `man nombre_de_orden`. Para salir del manual de ayuda en cualquier momento presione la tecla **q**. Haciendo uso del manual, resuelva los ejercicios propuestos en cada caso.

Por supuesto, también se puede invocar el manual del mismo **man**.



A continuación algunas ordenes básicas

### Mostrando la fecha y hora. La orden `date`

```
osboxes@osboxes:~$ date
Mon 21 Mar 2022 01:40:56 PM -05
osboxes@osboxes:~$
```

`date` imprime el día de la semana, la fecha y la hora actual del sistema (reloj de 24 horas, la zona de tiempo según la configuración del sistema).

### Ejercicio

- 1.- Haciendo uso de la orden `date` muestre la fecha de su cumpleaños. Si su cumpleaños es el 27 de Octubre, la orden a `date` se le debe pasar “27 Oct”, mediante la opción apropiada .

### Mostrando el calendario. La orden `cal`

```
osboxes@osboxes:~$ cal
      March 2022
Su Mo Tu We Th Fr Sa
                1  2  3  4  5
 6  7  8  9 10 11 12
13 14 15 16 17 18 19
20 21 22 23 24 25 26
27 28 29 30 31
osboxes@osboxes:~$
```

Ejercicios. Muestre por pantalla:

- 1.- El calendario de todos los meses del presente año.
- 2.- Los meses transcurridos desde el inicio de año hasta el presente mes.

### Haciendo eco de caracteres. La orden `echo`

La orden `echo` imprime (o hace eco) en la terminal cualquier cosa que se escriba en la línea. (hay algunas excepciones que se mencionarán después):

```
osboxes@osboxes:~$ echo Este es mi primer laboratorio preliminar de SO
Este es mi primer laboratorio preliminar de SO
osboxes@osboxes:~$ echo

osboxes@osboxes:~$ echo uno      dos      tres
uno dos tres
osboxes@osboxes:~$
```

En el último ejemplo se puede observar que `echo` extrae los espacios en blanco adicionales entre palabras. Eso es porque en un sistema *Unix like*, las palabras son importantes; los espacios en blanco están meramente allí para separar las palabras, generalmente el sistema ignora los espacios en blanco extras.

### Ejercicios

- 1.- Modifique el último ejemplo de forma que `echo` no ignore los espacios en blanco.
- 2.- Haciendo uso de la orden `echo` muestre por pantalla *hola como estas*, donde cada palabra debe estar en filas distintas pero en la misma columna.

## Trabajando con directorios

El directorio en el que se encuentra después de haberse identificado, se denomina *home*. En este directorio usted tiene derecho de crear, copiar, borrar, renombrar o mover directorios y archivos. Si usted en algún momento necesita saber cuál es su *home*, puede imprimir la variable de entorno `HOME`, anteponiéndole el signo `:`:

```
osboxes@osboxes:~$ echo $HOME
/home/osboxes
osboxes@osboxes:~$
```

Para crear directorios utilizamos la orden `mkdir`. Por ejemplo si deseamos crear tres directorio en nuestro *home*, utilizaremos la siguiente línea.

```
osboxes@osboxes:~$ ls
Desktop Documents Downloads Music Pictures Public Templates Videos
osboxes@osboxes:~$ mkdir dir1 dir2 dir3
osboxes@osboxes:~$ ls
Desktop dir2 Documents Music Public Videos
dir1 dir3 Downloads Pictures Templates
osboxes@osboxes:~$
```

En este caso no se obtiene mensaje alguno tras ejecutarse la orden. Una de las formas de ver si se han creado los directorios, es listando su contenido. Esto lo logramos con la orden `ls`, que después será explicado con más detalle.

Se puede crear subdirectorios intermedios en una sola orden, sin cambiar de directorio.

```
osboxes@osboxes:~$ mkdir -p dir1/dir1.1/dir1.1.1
osboxes@osboxes:~$
```

Crearé el directorio `dir1.1.1` después de haber creado `dir1.1` y por debajo de este.

Se denomina directorio actual, o directorio de trabajo, al directorio que se encuentra en el momento en que se ejecuta una orden. Al iniciar una sesión el directorio actual es su *home*. Para saber cuál es el directorio de trabajo use la orden `pwd`.

```
osboxes@osboxes:~$ pwd
/home/osboxes
osboxes@osboxes:~$
```

Para cambiar de un directorio a otro, empleamos la orden `cd`. Por ejemplo para ingresar al directorio de mayor profundidad desde el *home*.

```
osboxes@osboxes:~$ cd dir1/dir1.1/dir1.1.1/
osboxes@osboxes:~/dir1/dir1.1/dir1.1.1$ pwd
/home/osboxes/dir1/dir1.1/dir1.1.1
osboxes@osboxes:~/dir1/dir1.1/dir1.1.1$
```

Para regresar al directorio *home* desde cualquier directorio no es necesario indicar toda la ruta, simplemente indicar la orden: `cd`. Para regresar al directorio previo donde nos encontramos podemos hacerlo del siguiente modo `cd -`

```
osboxes@osboxes:~/dir1/dir1.1/dir1.1.1$ cd
osboxes@osboxes:~$ pwd
/home/osboxes
osboxes@osboxes:~$ cd -
/home/osboxes/dir1/dir1.1/dir1.1.1
osboxes@osboxes:~/dir1/dir1.1/dir1.1.1$ pwd
/home/osboxes/dir1/dir1.1/dir1.1.1
osboxes@osboxes:~/dir1/dir1.1/dir1.1.1$
```

Como puede verse, `cd -` puede usarse para moverse entre dos directorios.

En los ejemplos arriba mostrados se han indicado rutas tales como `dir1/dir1.1/dir1.1.1` y `/home/osboxes/dir1`.

Cuando la ruta inicia con un *slash* (/) se dice que es una **ruta absoluta**, e indica que la ruta debe iniciarse desde el directorio raíz. En caso contrario se denomina **ruta relativa** y se inicia en el directorio actual. Se denomina directorio padre al directorio que se encuentra por encima del directorio actual. Tanto el directorio actual como el directorio padre pueden ser representados por dos símbolos:

- . directorio actual
- .. directorio padre

Estos símbolos pueden ser usados en ordenes como `cd`. Por ejemplo para regresar al directorio padre.

```
osboxes@osboxes:~/dir1/dir1.1/dir1.1.1$ pwd
/home/osboxes/dir1/dir1.1/dir1.1.1
osboxes@osboxes:~/dir1/dir1.1/dir1.1.1$ cd ..
osboxes@osboxes:~/dir1/dir1.1$ pwd
/home/osboxes/dir1/dir1.1
osboxes@osboxes:~/dir1/dir1.1$
```

Debe respetar el espacio entre `cd` y el punto-punto (`..`)

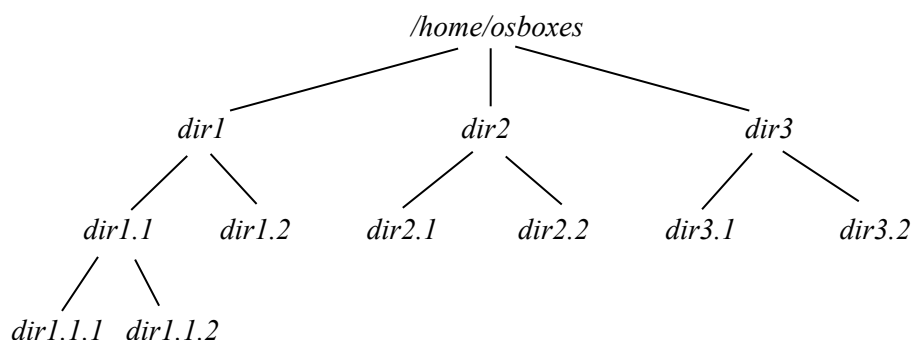
Es importante mencionar que el símbolo `~` representa al *home* en este caso en particular el directorio `/home/osboxes`. Esto puede abreviar la forma de nombrar las rutas, por ejemplo si se desea hacer referencia al directorio `/home/osboxes/dir1/dir1.1` se puede emplear `~/dir1/dir1.1`

Para borrar directorio vacíos empleamos la orden `rmdir`.

```
osboxes@osboxes:~$ cd
osboxes@osboxes:~$ ls
Desktop  dir2  Documents  Music  Public  Videos
dir1     dir3  Downloads  Pictures  Templates
osboxes@osboxes:~$ mkdir fantasma
osboxes@osboxes:~$ ls
Desktop  dir2  Documents  fantasma  Pictures  Templates
dir1     dir3  Downloads  Music  Public  Videos
osboxes@osboxes:~$ rmdir fantasma
osboxes@osboxes:~$ ls
Desktop  dir2  Documents  Music  Public  Videos
dir1     dir3  Downloads  Pictures  Templates
osboxes@osboxes:~$
```

## Ejercicio

1.- Ejecute las ordenes necesarias para obtener el siguiente árbol de directorios:



2.- Haciendo uso de la orden `tree` compruebe que el árbol que ha creado en el ejercicio anterior, es el correcto. Previamente debe instalarlo con la orden: `sudo apt install tree`

3.- ¿Cuántas ejecuciones de la orden `rmdir`, como mínimo, son necesarias para borrar todo el árbol de directorios creados en el ejercicio anterior? Borre todos los directorios creados por debajo de `/home/osboxes`

## Trabajando con archivos

El sistema reconoce sólo tres tipos básicos de archivos: archivos ordinarios, archivos de directorios, y archivos especiales. Un archivo ordinario es sólo eso: cualquier archivo en el sistema que contiene datos, texto, instrucciones de un programa, o cualquier otra cosa. Los directorios no son sino archivos cuya estructura está definida de acuerdo al sistema operativo. Como se puede deducir de su nombre, un archivo especial tiene un significado especial para el sistema y típicamente está asociado con alguna forma de entrada/salida. Un nombre de archivo puede estar compuesto por cualquier carácter disponible desde el teclado (y aún los que no lo están) y el número de caracteres que contiene el nombre no debe ser mayor de 255 en caso contrario los caracteres extras son ignorados.

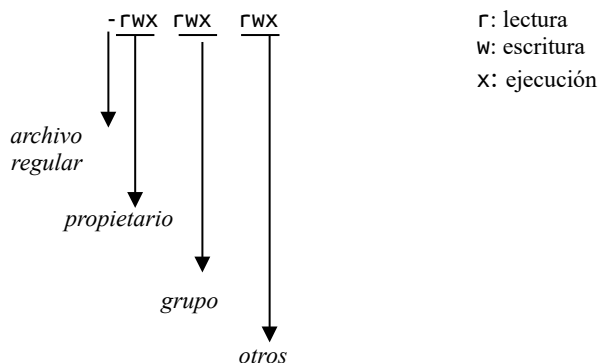
### Listando los archivos. La orden `ls`

```
osboxes@osboxes:~$ ls
Desktop Documents Downloads Music Pictures Public Templates Videos
osboxes@osboxes:~$ ls -l
total 32
drwxr-xr-x 2 osboxes osboxes 4096 Jan 15 09:25 Desktop
drwxr-xr-x 2 osboxes osboxes 4096 Jan 15 09:25 Documents
drwxr-xr-x 2 osboxes osboxes 4096 Jan 15 09:25 Downloads
drwxr-xr-x 2 osboxes osboxes 4096 Jan 15 09:25 Music
drwxr-xr-x 2 osboxes osboxes 4096 Jan 15 09:25 Pictures
drwxr-xr-x 2 osboxes osboxes 4096 Jan 15 09:25 Public
drwxr-xr-x 2 osboxes osboxes 4096 Jan 15 09:25 Templates
drwxr-xr-x 2 osboxes osboxes 4096 Jan 15 09:25 Videos
osboxes@osboxes:~$
```

El número a continuación de la palabra `total`, al inicio de la lista en el segundo ejemplo, no corresponde al número de archivos como se podría pensar, sino al número total de KB que se ha asignado a los archivos listados en el directorio (no confundir con el tamaño del archivo). El sistema asigna zonas para almacenar los archivos, cada zona es de 4KB. La asignación de zonas se hace redondeando al mayor entero el resultado de dividir el tamaño del archivo entre bloques de 4KB. Por ejemplo si el tamaño de un archivo es de 179 bytes, Linux le asigna una zona, es decir 4KB. Si el archivo es de 4097 bytes el sistema le asigna 2 zonas de 4KB cada una. Esta información sólo se muestra sólo cuando se lista todo el contenido de un directorio.

El **propietario** de un archivo es por lo general el usuario que lo ha creado. En el listado anterior la tercera columna indica que el propietario es el usuario `osboxes`. Todo usuario debe ser miembro de un **grupo**. En nuestro ejemplo la cuarta columna, indica que pertenece al grupo `osboxes`. Cualquier otro usuario que no pertenezca al grupo y que no sea el propietario se les denomina comúnmente *otros*.

Cuando en un listado largo de archivos (orden `ls -l`) aparece un guión en la primera columna, indica que se trata de un archivo regular. Las siguientes tres letras `rwX` indican que el propietario tiene permiso de leer, escribir y ejecutar si se trata de un archivo. Cuando un permiso le está negado, en lugar de alguna de las tres letras (`r`, `w`, `x`) se encuentra un guión. Las siguientes tres letras tienen el mismo significado pero corresponden a los permisos que tienen los miembros del grupo. Y las últimas tres letras corresponden a los permisos que se les otorga sobre el archivo a otros usuarios que no son ni el propietario ni miembro del grupo.





Por ejemplo

```
-rwxrw-r-- 1 osboxes osboxes 708 Mar 17 12:30 Makefile
```

El primer guión indica que se trata de un archivo regular. El propietario *osboxes* puede leer, escribir y ejecutar este archivo (siempre y cuando sea un archivo apropiado). Cualquier usuario que sea miembro del grupo *osboxes* puede leer y escribir sobre él. Pero cualquier otro usuario sólo podrán leerlo.

## Ejercicios

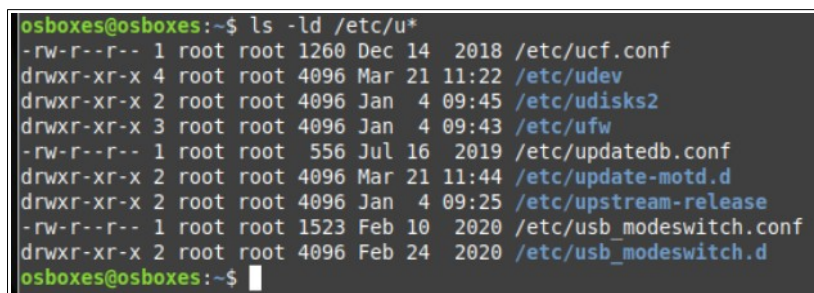
- 1.- Cuando se hace un listado largo la opción por defecto es listar el contenido del directorio ordenado alfabéticamente. Obtenga un listado pero en orden inverso.
- 2.- Si se desea listar en formato largo información del directorio */etc*, un primer intento sería la siguiente orden: `ls -l /etc`, sin embargo no se obtendría el resultado deseado, porque este mostraría información acerca de cada archivo que contiene el directorio */etc*. Escriba en la terminal la orden `ls` con la opción apropiada para mostrar sólo información del directorio */etc* en lugar de su contenido.
- 3.- Muestre el contenido del directorio */etc* en formato largo ordenando los archivos basándose en la última fecha de modificación de cada archivo.
- 4.- Lo mismo que el caso anterior, pero en orden inverso.
- 5.- Se desea mostrar los archivos contenidos en */usr/bin*, ordenados por tamaño.
- 6.- Liste en formato largo el contenido del directorio */etc* de forma que también se muestre el contenido de todos los directorios que se encuentran en */etc*. Si estos a su vez contienen directorios, también deben mostrarse.
- 7.- Busque información en Internet acerca de ¿qué es un inodo (o nodo-i)? Haciendo uso de la orden `ls` obtenga un listado de los archivos contenidos en */usr/bin*, con sus respectivos *inodos*.

## Sustitución de nombres de archivos (*Globbering*)

La sustitución de nombres de archivos (o *globbing*) es el método por el cual el *shell* expande una cadena conteniendo meta-caracteres o comodines (*wildcards*) en una lista de nombres de archivos. A continuación los comodines usados en *globbing*

- |                       |  |
|-----------------------|--|
| *                     | Coincide cero o más ocurrencias de cualquier carácter.           |
| ?                     | Coincide una ocurrencia de cualquier carácter.                   |
| [ <i>caracteres</i> ] | Coincide una ocurrencia de alguno de los <i>caracteres</i> dados |

Algunas ordenes o *scripts* que operan con archivos puede hacer uso de *globbing*. A continuación se presentarán algunos ejemplos usando la orden `ls` para ilustrar el resultado de *globbing*.



```
osboxes@osboxes:~$ ls -ld /etc/u*
-rw-r--r-- 1 root root 1260 Dec 14 2018 /etc/ucf.conf
drwxr-xr-x 4 root root 4096 Mar 21 11:22 /etc/udev
drwxr-xr-x 2 root root 4096 Jan 4 09:45 /etc/udisks2
drwxr-xr-x 3 root root 4096 Jan 4 09:43 /etc/ufw
-rw-r--r-- 1 root root 556 Jul 16 2019 /etc/updatedb.conf
drwxr-xr-x 2 root root 4096 Mar 21 11:44 /etc/update-motd.d
drwxr-xr-x 2 root root 4096 Jan 4 09:25 /etc/upstream-release
-rw-r--r-- 1 root root 1523 Feb 10 2020 /etc/usb_modeswitch.conf
drwxr-xr-x 2 root root 4096 Feb 24 2020 /etc/usb_modeswitch.d
osboxes@osboxes:~$
```

Se lista tanto archivos como directorios cuyo nombre inician con la letra *u*, en el directorio */etc*

```
osboxes@osboxes:~$ ls -d /etc/*.d
/etc/apparmor.d      /etc/modprobe.d      /etc/rcS.d
/etc/bash_completion.d /etc/modules-load.d  /etc/request-key.d
/etc/binfmt.d        /etc/pam.d           /etc/rsyslog.d
/etc/cron.d          /etc/profile.d       /etc/sane.d
/etc/depmod.d        /etc/rc0.d           /etc/sensors.d
/etc/environment.d   /etc/rc1.d           /etc/sudoers.d
/etc/grub.d          /etc/rc2.d           /etc/sysctl.d
/etc/init.d          /etc/rc3.d           /etc/tmpfiles.d
/etc/ld.so.conf.d    /etc/rc4.d           /etc/update-motd.d
/etc/libpaper.d      /etc/rc5.d           /etc/usb_modeswitch.d
/etc/logrotate.d     /etc/rc6.d
```

Se lista tanto archivos como directorios cuyo nombre terminan con la cadena *.d*, en el directorio */etc*  
En este caso solo hay nombres de directorios que coinciden con dicho patrón.

```
osboxes@osboxes:~$ ls -ld /etc/rc?.d
drwxr-xr-x 2 root root 4096 Jan 15 09:28 /etc/rc0.d
drwxr-xr-x 2 root root 4096 Jan 15 09:28 /etc/rc1.d
drwxr-xr-x 2 root root 4096 Jan 15 09:28 /etc/rc2.d
drwxr-xr-x 2 root root 4096 Jan 15 09:28 /etc/rc3.d
drwxr-xr-x 2 root root 4096 Jan 15 09:28 /etc/rc4.d
drwxr-xr-x 2 root root 4096 Jan 15 09:28 /etc/rc5.d
drwxr-xr-x 2 root root 4096 Jan 15 09:28 /etc/rc6.d
osboxes@osboxes:~$
```

Se lista tanto archivos como directorios cuyo nombre sea igual a *recX.d* donde *X* puede ser cualquier carácter. Observe que antes del punto debe de haber exactamente 4 caracteres.

```
osboxes@osboxes:~$ ls -ld /etc/rc[0-9].d
drwxr-xr-x 2 root root 4096 Jan 15 09:28 /etc/rc0.d
drwxr-xr-x 2 root root 4096 Jan 15 09:28 /etc/rc1.d
drwxr-xr-x 2 root root 4096 Jan 15 09:28 /etc/rc2.d
drwxr-xr-x 2 root root 4096 Jan 15 09:28 /etc/rc3.d
drwxr-xr-x 2 root root 4096 Jan 15 09:28 /etc/rc4.d
drwxr-xr-x 2 root root 4096 Jan 15 09:28 /etc/rc5.d
drwxr-xr-x 2 root root 4096 Jan 15 09:28 /etc/rc6.d
osboxes@osboxes:~$
```

Se lista tanto archivos como directorios cuyo nombre sea igual a *recX.d* donde *X* tiene que ser uno de los dígitos entre 0 y 9 . Observe que la salida es ligeramente distinta a la anterior.

```
osboxes@osboxes:~$ ls -ld /etc/rc[!0-9].d
drwxr-xr-x 2 root root 4096 Jan 4 09:46 /etc/rcS.d
osboxes@osboxes:~$
```

Se lista tanto archivos como directorios cuyo nombre sea igual a *recX.d* donde *X* **no** tiene que ser uno de los dígitos entre 0 y 9 . Observe que la salida es lo opuesto de la anterior

## Buscando archivos por su nombre u otro criterio. La orden *find*

Esta orden busca archivos, dado un criterio, en un árbol de directorio. El directorio a partir del cual se hace la búsqueda debe ser proporcionado. En caso que no se proporcione alguno, se asume el directorio actual. El criterio más común para hacer búsqueda es el nombre. Pero usted puede hacer búsqueda por fechas, por tamaño de archivo, por tipo de archivo, etc. Adicionalmente usted puede indicar que se ejecute alguna orden sobre los archivos encontrados.

La búsqueda sobre los directorios está supeditada a los permisos que tenga cada directorio. Es decir si en el árbol de directorios hay algún directorio al que usted no tiene permiso de acceder, se mostrará el siguiente mensaje de error: Permiso denegado.

También puede hacer uso de comodines (*globbing*).

Por ejemplo

```
find / -name *.c
```

Busca todos los archivos que tengan extensión *.c* en su dispositivo de almacenamiento. Es probable que usted obtenga una lista larga como salida.

Ejercicios. Emplee la orden *find* para atender las siguientes solicitudes:

- 1.- Busque en el directorio */etc* todos los archivos mayores a 10 KB.
- 2.- Busque en su disco duro archivos vacíos.
- 3.- Busque en su disco duro los archivos con extensión *.c* menores de 10KB.
- 4.- Busque en su disco el archivo más reciente creado.
- 5.- Elimine todos los directorios vacíos que se encuentre en su directorio *home*.

### Mostrando el estado de un archivo. La orden *stat*

```
osboxes@osboxes:~$ stat /etc/passwd
  File: /etc/passwd
  Size: 2748          Blocks: 8          IO Block: 4096   regular file
Device: 801h/2049d   Inode: 7602403       Links: 1
Access: (0644/-rw-r--r--)  Uid: (  0/   root)   Gid: (  0/   root)
Access: 2022-03-21 10:45:41.168000000 -0500
Modify: 2022-01-15 09:21:22.848020348 -0500
Change: 2022-01-15 09:21:22.852020348 -0500
 Birth: -
osboxes@osboxes:~$ stat /usr/bin/gnome-terminal
  File: /usr/bin/gnome-terminal
  Size: 3400          Blocks: 8          IO Block: 4096   regular file
Device: 801h/2049d   Inode: 11272797      Links: 1
Access: (0755/-rwxr-xr-x)  Uid: (  0/   root)   Gid: (  0/   root)
Access: 2022-03-21 10:45:21.901565414 -0500
Modify: 2020-12-04 06:50:16.000000000 -0500
Change: 2022-01-15 09:18:05.348012395 -0500
 Birth: -
osboxes@osboxes:~$
```

### Ejercicios

1.- Haciendo uso de la orden *stat* obtenga por pantalla las siguientes salidas:

- a) Nombre de archivo: */usr/bin/haddock*  
Tipo de archivo: regular file
- b) Número de bloques lógicos asignados: 69232  
Tamaño de cada bloque lógico: 512

### Concatenando archivos. La orden *cat*

```
osboxes@osboxes:~$ cat /usr/share/applications/python3.8.desktop
[Desktop Entry]
Name=Python (v3.8)
Comment=Python Interpreter (v3.8)
Exec=/usr/bin/python3.8
Icon=/usr/share/pixmaps/python3.8.xpm
Terminal=true
Type=Application
Categories=Development;
StartupNotify=true
NoDisplay=true
osboxes@osboxes:~$
```

Concatena varios archivos y los muestra por la salida estándar. Si solo se proporciona el nombre de un solo archivo, el contenido de este es mostrado por la salida estándar. Los archivos tipo texto (aquellos que contienen caracteres imprimibles) se mostrarán tal cual, mientras que para los archivos binarios, se mostrará caracteres extraños.



## Ejercicios

- 1.- Obtenga la salida del ejemplo anterior, pero ahora mostrando un número de línea en el lado izquierdo.
- 2.- Muestre el contenido de todos los archivos con extensión *.sh* que se encuentran en */usr/bin*

### Contando el número de palabras de un archivo. La orden *wc*

```
osboxes@osboxes:~$ wc /usr/share/applications/python3.8.desktop
10 14 220 /usr/share/applications/python3.8.desktop
osboxes@osboxes:~$
```

La orden *wc* lista tres números seguidos por el nombre. El primer número representa el número de líneas, el segundo es el número de palabras y el tercero es el número de caracteres contenidos en el archivo.

## Ejercicios

- 1.- Muestre por pantalla solo el número de líneas del archivo */etc/group*
- 2.- También muestre la longitud de la línea más larga, del mismo archivo.

### Entrada, salida y error estándar

Cada vez que el *shell* ejecuta un programa, prepara para éste los dispositivos de entrada, salida y error estándar. Por defecto el dispositivo asociado a la entrada estándar es el teclado y para la salida y error estándar es la pantalla. De esta forma el programa ejecutable (proceso) tiene un medio de comunicación con el mundo exterior. El dispositivo correspondiente al teclado y a la pantalla se llama */dev/ttyc1*, el de la impresora es */dev/lp*, el del puerto serial es */dev/tty00*. También existe un dispositivo ficticio de salida que recibe los datos y los desecha. Este dispositivo se llama */dev/null*. Muchas veces se desea que los resultados de un programa sean guardados en un archivo en lugar de imprimirlos por pantalla, otras veces se desea ingresar los datos desde un archivo en lugar de ingresarlos desde el teclado. Todo ello por supuesto sin tener que modificar el programa. El *shell* acepta algunos símbolos que permiten cambiar el dispositivo estándar para un programa en ejecución, redirigiendo el flujo de I/O. Por ejemplo, con la siguiente orden:

```
osboxes@osboxes:~$ ls -l /etc > milista
osboxes@osboxes:~$
```

Debido al símbolo *>* se cambia la salida estándar para la orden *ls*, redirigiéndola hacia el archivo *milista*. Por consiguiente los resultados de la orden *ls* en vez de aparecer por pantalla serán enviados a un archivo creado en ese momento, si ya existiese se sobrescribe. Para que la salida sea añadida en lugar de sobre escribirla, use el símbolo *>>* colocando a continuación el nombre del archivo.

También se puede redirigir la salida de una lista de ordenes:

```
osboxes@osboxes:~$ (date;uptime;df) > milog
osboxes@osboxes:~$ cat milog
Mon 21 Mar 2022 04:31:58 PM -05
16:31:58 up 5:48, 1 user, load average: 0.07, 0.04, 0.00
Filesystem      1K-blocks      Used Available Use% Mounted on
udev              1968108          0   1968108   0% /dev
tmpfs             403012        1160    401852   1% /run
/dev/sda1        226639372 8481876 206575188   4% /
tmpfs             2015056          0   2015056   0% /dev/shm
tmpfs              5120           4      5116   1% /run/lock
tmpfs             2015056          0   2015056   0% /sys/fs/cgroup
/dev/sda3        277707844 128264 263403120   1% /home
tmpfs             403008         36    402972   1% /run/user/1000
osboxes@osboxes:~$
```

El símbolo < redirige la entrada estándar, por ejemplo

```
osboxes@osboxes:~$ wc < milista
244  2201 13474
osboxes@osboxes:~$
```

presenta por pantalla la cantidad de líneas que contiene el archivo *milista*

Ahora suponga que no se desea redirigir la entrada desde un archivo, sino desde una secuencia de líneas de caracteres que se desean ingresar en ese mismo momento, sin necesidad de que esta secuencia se encuentre en un archivo. Para este caso usted puede usar << *delimitador*. Esto ocasionará que el *shell* muestre el *prompt* secundario en espera a que usted ingrese la secuencia de línea de caracteres. Cada vez que desee una línea presione Enter. La última línea debe ser el *delimitador*.

Por ejemplo

```
osboxes@osboxes:~$ wc -l << END
>
osboxes@osboxes:~$ wc -l << END
> (1) Primera línea
> (2) Segunda línea
> (3) Tercera línea
> END
3
osboxes@osboxes:~$
```

Callouts:

- Presionar Enter
- Prompt secundario en espera del ingreso de líneas
- Usted debe ingresar las líneas. La última debe ser END

Como se puede apreciar todas las líneas fueron tomadas como un texto e ingresado como entrada estándar a la orden *wc*, mostrando el número de líneas de este texto temporal.

El símbolo de canalización | crea una tubería (*pipe*) entre varios programas que se ejecutan en forma concurrente. Veamos qué es lo que hace la siguiente orden.

```
osboxes@osboxes:~$ du -a /usr/bin | sort -nr | more
165748 /usr/bin
8244 /usr/bin/gdb
5364 /usr/bin/python3.8
4472 /usr/bin/aptitude-curses
3400 /usr/bin/perl5.30.0
3400 /usr/bin/perl
2380 /usr/bin/Xephyr
2276 /usr/bin/Xwayland
2260 /usr/bin/x86_64-linux-gnu-ld.gold
2124 /usr/bin/busybox
2000 /usr/bin/x86_64-linux-gnu-dwp
1820 /usr/bin/nemo
1804 /usr/bin/nemo-desktop
1700 /usr/bin/x86_64-linux-gnu-ld.bfd
1616 /usr/bin/pnm2ppa
1552 /usr/bin/systemd-analyze
1548 /usr/bin/strace
1380 /usr/bin/flatpak
1348 /usr/bin/rpcclient
1320 /usr/bin/vim.tiny
1176 /usr/bin/seahorse
1156 /usr/bin/bash
1140 /usr/bin/gnome-keyring-daemon
--More--
```

La salida de la orden `du` (*disk usage*) muestra el espacio ocupado por cada archivo (en sectores), contenido en el directorio especificado (directorio de trabajo por defecto). Esta salida se pasa como entrada para la orden `sort` quien ordena esta lista por tamaños (primera columna de la salida de la orden `du`) de mayor a menor (considera la columna como números). Este a su vez lo pasa a `more` para hacer una pausa en caso de que sea necesario.

Puede limpiar su pantalla con la orden `clear`.

Para redirigir la salida a un archivo y a la salida estándar (pantalla), se puede usar la orden `tee`

```
osboxes@osboxes:~$ date | tee fsalida
Mon 21 Mar 2022 04:42:00 PM -05
osboxes@osboxes:~$ cat fsalida
Mon 21 Mar 2022 04:42:00 PM -05
osboxes@osboxes:~$
```

## Ejercicios

- 1.- Haciendo uso de tuberías muestre ¿cuántos archivos con extensión `.sh` existen en `/usr/bin`?
- 2.- Concatene todos los archivos hallados en el ejercicio anterior en uno solo, con nombre `lscripts` en el directorio actual.

## Identificando el tipo de archivo. La orden `file`

```
osboxes@osboxes:~$ file /usr/bin/yes
/usr/bin/yes: ELF 64-bit LSB shared object, x86-64, version 1 (SYSV), dynamically linked, interpreter /lib64/ld-linux-x86-64.so.2, BuildID[sha1]=073449e3e04dc523b2077be9a03685a970842371, for GNU/Linux 3.2.0, stripped
osboxes@osboxes:~$ file /usr/local/bin/search
/usr/local/bin/search: Bourne-Again shell script, ASCII text executable
osboxes@osboxes:~$ file /usr/share/pixmaps/gvim.svg
/usr/share/pixmaps/gvim.svg: SVG Scalable Vector Graphics image
osboxes@osboxes:~$ file /usr/share/pixmaps/debian-logo.png
/usr/share/pixmaps/debian-logo.png: PNG image data, 48 x 48, 8-bit colormap, non-interlaced
osboxes@osboxes:~$ file /usr/share/applications/python3.8.desktop
/usr/share/applications/python3.8.desktop: ASCII text
osboxes@osboxes:~$
```

## Ejercicios

- 1.- Todos los ejemplos de la orden `file` se pudieron escribir en una sola línea. ¿Cuál es esta?
- 2.- Empleando la orden `file` muestre información de cada una de las particiones del disco duro.

## Haciendo una copia de un archivo. La orden `cp`

Si se desea copiar archivos, entonces deberá emplear la orden `cp`. Por ejemplo, para realizar una copia del archivo `milista` con el nombre `milista.old`

```
osboxes@osboxes:~$ ls
Desktop  Downloads  milista  Music  Public  Videos
Documents  fsalida  milog  Pictures  Templates
osboxes@osboxes:~$ cp milista milista.old
osboxes@osboxes:~$ ls
Desktop  Downloads  milista  milog  Pictures  Templates
Documents  fsalida  milista.old  Music  Public  Videos
osboxes@osboxes:~$
```

El formato de la orden `cp` con algunas de sus opciones se indica a continuación::

```
cp [-fip] fuelle destino
cp [-fipR] fuelle1 fuente2 . . . dir_destino
```

La primera forma se aplica cuando el destino es un archivo, en este caso el origen debe ser también un archivo. La segunda forma se aplica cuando el destino es un directorio y el origen puede ser un archivo o conjunto de archivos, indicado inclusive por comodines.

En cuánto a las opciones podemos mencionar las siguientes:

-i Pregunta de modo iterativo si el archivo se desea sobrescribir, obviamente cuando el archivo que se va a copiar ya existe, en caso contrario no tiene efecto.

-r -R Se copian los archivos y los subdirectorios recursivamente contenidos en el directorio fuente. En este caso el destino tiene que ser un directorio.

Ejercicios

- 1.- Copie a su *home* todos los archivos con extensión *.h* que se encuentran en */usr/include*
- 2.- Copie el contenido completo del directorio */etc* dentro del directorio *Descargas* de su *home*.

### Renombrando un archivo. La orden `mv`

Un archivo puede ser renombrado con la orden `mv` y a continuación el nombre actual seguido por el nuevo nombre del archivo.

```
osboxes@osboxes:~$ ls
Desktop  Downloads  milista      milog  Pictures  Templates
Documents fsalida    milista.old  Music  Public    Videos
osboxes@osboxes:~$ mv milista.old mi_lista.old
osboxes@osboxes:~$ ls
Desktop  Downloads  milista      milog  Pictures  Templates
Documents fsalida    mi_lista.old Music  Public    Videos
osboxes@osboxes:~$
```

En el ejemplo anterior la orden `mv` se usó para cambiar el nombre de un archivo. Sin embargo, cuando los dos argumentos para esta orden hace referencia a directorios diferentes, entonces el archivo es movido del primer directorio al segundo.

Ejercicios

- 1.- Cambie la extensión a todos los archivos que fueron copiados en el apartado anterior (con extensión *.h*), por el de *.bak*
- 2.- Mueva los archivos que ha renombrado en el paso anterior, al directorio *Descargas*

### Removiendo un archivo. La orden `rm`

Para eliminar o remover un archivo del sistema, emplee la orden `rm`. El argumento para `rm` es simplemente el nombre del archivo a ser removido.

```
osboxes@osboxes:~$ ls
Desktop  Downloads  milista      milog  Pictures  Templates
Documents fsalida    mi_lista.old Music  Public    Videos
osboxes@osboxes:~$ rm mi_lista.old
osboxes@osboxes:~$ ls
Desktop  Downloads  milista      Music  Public    Videos
Documents fsalida    milog        Pictures  Templates
osboxes@osboxes:~$
```

## Ejercicios

- 1.- Borre de forma imperativa (con una sola orden) todos los archivos que inicien con la letra *t* y que se encuentran en el directorio *Descargas* de su *home*.
- 2.- Ejecute la orden *rm* de forma que le pregunte uno a uno qué archivo desea borrar. Borre de forma alternada los archivos, cuando el *shell* le pregunte.
- 3.- En el directorio *Descargas*, borre el directorio *etc* y todo lo que contiene.

### Cambiando los permisos a un archivo. La orden *chmod*

A diferencia de MS-Windows el hecho de que un archivo tenga una extensión *.com* o *.exe* no significa que sea un programa ejecutable. La acción de ejecutar cualquier programa esta condicionada al permiso correspondiente *x* de ejecución del archivo. Esto es importante a la hora de escribir programas, como los *scripts*. Un *script* es semejante a un archivo por lotes (*bat*) del DOS. Estos archivos son de tipo texto, y para que se ejecuten se le deberá colocar el permiso de ejecución con la orden *chmod*.

La orden *chmod*, cambia los permisos de acceso sobre archivos, sólo el propietario puede modificarlos. El formato es el siguiente:

```
chmod a|u|g|o +/- r|w|x nombre_de_archivo
```

La letra *a* (*all*) indica que el permiso se aplicará a todos: propietario, grupos u otros. Si se desea modificar el permiso solo al propietario se tendrá que escribir *u* (*user*) en lugar de *a*. En los demás casos emplee la letra *g* para grupo y *o* para otros. El carácter *-* indica que se le está quitando los permisos, si desea añadirle en lugar de quitarle, escriba *+* en lugar de *-*. Y por último el siguiente carácter deberá ser: *r*, *w* ó *x* que indican el tipo de permiso que deseamos eliminar o agregar. A continuación primero quitaremos el permiso de escritura al archivo *milista* para todos (*all*) los usuarios, para luego restituirla:

```
osboxes@osboxes:~$ ls -l milista
-rw-rw-r-- 1 osboxes osboxes 13474 Mar 21 16:28 milista
osboxes@osboxes:~$ chmod a-w milista
osboxes@osboxes:~$ ls -l milista
-r--r--r-- 1 osboxes osboxes 13474 Mar 21 16:28 milista
osboxes@osboxes:~$ ls -l >> milista
bash: milista: Permission denied
osboxes@osboxes:~$ chmod a+w milista
osboxes@osboxes:~$ ls -l milista
-rw-rw-rw- 1 osboxes osboxes 13474 Mar 21 16:28 milista
osboxes@osboxes:~$ ls -l >> milista
osboxes@osboxes:~$ ls -l milista
-rw-rw-rw- 1 osboxes osboxes 14100 Mar 21 16:57 milista
osboxes@osboxes:~$
```

## Ejercicios

- 1.- Cambie los permisos de todos los archivos contenidos en *~/Descargas* de forma que cualquier usuario pueda modificarlos.
2. Cree un directorio con nombre *foo*, elimine el permiso de escritura para todos los casos al directorio *foo*. Ahora copie el archivo */etc/passwd* al directorio *foo*. ¿Qué sucede? Asigne el permiso de escritura para todos los usuarios, vuelva a copiar el archivo. ¿Qué sucede?
- 3.- Al directorio *foo* creando en el ejercicio anterior elimine el permiso de ejecución. Ingrese a dicho directorio. ¿Qué sucede?
- 4.- Con la ayuda de Internet averigüe sobre el permiso SUID. ¿Qué archivos tienen este permiso en el directorio */usr/bin*?
- 5.- Con la ayuda de Internet averigüe sobre el permiso SGID. ¿Qué archivos tienen este permiso en



el directorio `/usr/bin`?

6.- Con la ayuda de Internet averigüe sobre el permiso denominado *sticky bit*. ¿Qué directorios tienen este permiso en el disco duro?

### Creando enlaces entre archivos. La orden `ln`

```
osboxes@osboxes:~$ ls -l milista
-rw-rw-rw- 1 osboxes osboxes 14100 Mar 21 16:57 milista
osboxes@osboxes:~$ ln milista mi_lista
osboxes@osboxes:~$ ls -li milista mi_lista
7602753 -rw-rw-rw- 2 osboxes osboxes 14100 Mar 21 16:57 mi_lista
7602753 -rw-rw-rw- 2 osboxes osboxes 14100 Mar 21 16:57 milista
osboxes@osboxes:~$
```

Observe que tiene el mismo número de *inodo*

Se ha creado un *hard link* entre *milista* y *mi\_lista*. En este caso se crea una nueva entrada de directorio con el nombre *mi\_lista* en el directorio de trabajo, pero el número de *inodo* es el mismo que el de *milista*. Un *inodo* es una estructura en el que se guarda información acerca del archivo (excepto su nombre), datos como los permisos, tipo de archivo, fecha de creación, fecha de modificación, tamaño, etc. Son guardados en el *inodo*. Todo archivo tiene un *inodo*. El *inodo* es manejado por el sistema operativo y no es accesible al usuario de forma directa. Se puede decir que un enlace es un nuevo nombre para el mismo archivo. Normalmente se emplea para que diferentes usuarios tengan acceso al mismo archivo, sin tener que duplicar la información.

Existen otro tipo de enlaces llamados *soft-link*. A diferencia del primero este enlaza dos archivos no mediante su *inodo*, sino a través de un archivo que contiene la ruta y el nombre del archivo enlazado.

```
osboxes@osboxes:~$ ln -s /etc/passwd
osboxes@osboxes:~$ ls -l
total 72
drwxr-xr-x 2 osboxes osboxes 4096 Jan 15 09:25 Desktop
drwxr-xr-x 2 osboxes osboxes 4096 Jan 15 09:25 Documents
drwxr-xr-x 2 osboxes osboxes 4096 Jan 15 09:25 Downloads
-rw-rw-r-- 1 osboxes osboxes 32 Mar 21 16:42 fsalida
-rw-rw-rw- 2 osboxes osboxes 14100 Mar 21 16:57 mi_lista
-rw-rw-rw- 2 osboxes osboxes 14100 Mar 21 16:57 milista
-rw-rw-r-- 1 osboxes osboxes 603 Mar 21 16:31 milog
drwxr-xr-x 2 osboxes osboxes 4096 Jan 15 09:25 Music
lrwxrwxrwx 1 osboxes osboxes 11 Mar 21 17:01 passwd -> /etc/passwd
drwxr-xr-x 2 osboxes osboxes 4096 Jan 15 09:25 Pictures
drwxr-xr-x 2 osboxes osboxes 4096 Jan 15 09:25 Public
drwxr-xr-x 2 osboxes osboxes 4096 Jan 15 09:25 Templates
drwxr-xr-x 2 osboxes osboxes 4096 Jan 15 09:25 Videos
osboxes@osboxes:~$
```


El archivo *passwd* es un enlace al archivo `/etc/passwd`. En este caso *passwd* y `/etc/passwd` son dos archivos completamente distintos. El primero contiene la ruta donde se encuentra el segundo, sin embargo cada vez que usted accede al archivo *passwd* el sistema se comporta como si hubiera accedido al archivo `/etc/passwd`.

### Archivos de configuración.

Existen algunos archivos que permiten configurar la forma de trabajar en la terminal. El nombre de estos archivos normalmente inician con un punto ('.'). El sistema por defecto no los muestra, ni con el navegador de archivos (en forma gráfica) ni en la terminal (con la orden `ls`). Para poder verlos se deben de listar con la opción `-a`, o en el navegador presionando `Ctrl+h`.

```
osboxes@osboxes:~$ ls -a
.      .cache      Documents  .gtkrc-xfce  milog        Public      .Xauthority
..     .cinnamon  Downloads  .linuxmint   Music        .sudo_as_admin_successful  .xsession-errors
.bash_history  .config    fsalida    .local       passwd       Templates   .xsession-errors.old
.bash_logout  .Desktop  .gnupg     mi_lista    Pictures     .themes
.bashrc       .dmrc     .gtkrc-2.0 milista     .profile    Videos
osboxes@osboxes:~$
```

Observe que son muchos archivos o directorios que inician con punto. Esto se debe por que muchos programas emplean archivos de configuración y con el uso de nuevos programas, estos archivos se incrementan.

**Importante:** cierre todas las terminales virtuales (  ) en las que ha estado trabajando.

Vuelva a abrir una terminal y copie el archivo *.bash\_history* con el de nombre *historial.txt*.

Además ejecute la siguiente orden:

```
lsb_release -a > mi_release.txt
```

A continuación suba los archivos *historial.txt* y *mi\_release.txt* al **Buzón** creado en Paideia.

*Prof. Alejandro T. Bello Ruiz*