

COMANDOS UBUNTU

Desde un terminal:

(modo background):

Para lanzar procesos en modo background, se añade & al final. Por ejemplo, abrir un editor de textos sin bloquear el terminal.

`gedit miFichero.cpp &`

Devuelve el identificador de proceso (PID). Este se puede ver con el comando `ps`.

Redirección de la salida:

< `fich1` La entrada estándar es el contenido del fichero `fich1`. `ls > fich1`

> `fich2` La salida estándar es el fichero `fich2`. `wc < fich2`

Pipes Comando1 | Comando2 | ... | comandoN: La salida de un comando y la entrada al siguiente se puede enlazar con pipes (tuberías).

`cat fichero1 | grep hola`

cat

Visualizar el contenido de un fichero.

`cat fichero_de_texto`. Se pueden indicar varios ficheros: `cat f1 f2 f3`

cd

Cambiar de directorio.

`cd ..` Sube al directorio padre.

`cd directorio` Cambia al directorio. Tiene que ser visible desde donde estamos.

`cd` Nos lleva al directorio home.

`.` Un punto representa el directorio actual.

`..` Doble punto el directorio padre.

Utilizar **tabulador** para completar las rutas.

Por ejemplo, si en el directorio actual hay un directorio llamado: curso

`cd c <<tab>>`

chmod

Para cambiar los permisos de un fichero. Se indican con grupos de 3. Ver comando `ls -l`

`rwX` Los 3 primeros para el dueño.

`rwX` Usuarios del mismo grupo.

`rwX` Otros usuarios.

Los 3 números se interpretan en binario: 110 100 100 → indica: rw-r--r--

`chmod 644 fichero1`

clear

Limpiar la pantalla.

cmp

Compara dos ficheros octeto a octeto. Si difiere lo señala, y en ese momento no analiza mas. Para marcar las diferencias, mejor el comando **diff**

`cmp f1 f2`

cp

Copia un fichero.

`cp fichero_origen fichero_destino`

Se pueden utilizar comodines / rutas de directorios: . .. *

date

Muestra la hora y la fecha del sistema.

diff

Indica las diferencias entre los dos ficheros: las diferencias del primero se muestran con < y las del segundo con >.

`diff f1 f2`

du

Muestra los directorios y lo que ocupan en bloques a partir de un directorio.

`du subdirectorio`

find

Imprime todo el subárbol del directorio indicado. Si no se indica directorio, muestra todo lo que cuelgue del directorio actual. Comando similar: **du**

`find subdirectorio`

free

Sirve para mostrar la memoria Ram disponible.

`free -h`

gcc

Compilador de C.

Para un fichero: `gcc miFichero.c -o nombre_ejecutable`.

Si omitimos el parámetro -o con el nombre_ejecutable. El nombre del fichero generado se llamará a.out.

Los ficheros .h no alteran ni se indican en el comando gcc.

Cuando queramos compilar incluyendo la librería math hay que compilar de la siguiente manera: `gcc fuente.c -o ejecutable -lm`

Cuando tenemos varios ficheros fuentes que compilar:

`gcc fuente1.c fuente2.c fuente3.c -o ejecutable`

`gcc *.c -o ejecutable` Todos los archivos C de la carpeta actual.

Para compilar con la librería de **Threads de POSIX**. Hay que indicar la librería de los threads para que entre en el proceso de compilación.

`gcc fuente.c -o ejecutable -pthread`

g++

Compilador de C++.

Para un fichero: `g++ miFichero.c -o nombre_ejecutable`.

Si omitimos el parámetro -o con el nombre_ejecutable. El nombre del fichero generado se llamará a.out.

Los ficheros .h no alteran ni se indican en el comando g++.

Cuando tenemos varios ficheros fuentes que compilar:

```
g++ fuente1.c fuente2.c fuente3.c -o ejecutable
```

```
g++ *.cpp -o ejecutable
```

 Todos los archivos CPP de la carpeta actual.

Compilar con otras versiones:

```
g++ -std=c++11 fichero.cpp -o ejecutable
```

```
g++ -std=c++14 fichero.cpp -o ejecutable
```

```
g++ -std=c++17 fichero.cpp -o ejecutable
```

gedit

Abrir un editor de textos. Colorea la sintaxis para C / C++.

```
gedit miFichero.cpp
```

grep

Para buscar texto dentro del contenido de uno o varios ficheros.

Mostrar las líneas de todos los ficheros cpp del directorio actual que contienen la palabra include:

```
grep include *.cpp
```

head

Muestra las primeras líneas de un fichero. Por defecto, son 10. Se puede indicar el número de líneas a mostrar.

Las 5 primeras:

```
head -5 fichero_texto
```

join

Dados dos ficheros, hace las combinaciones:

```
join f1 f2
```

```
a      1
b      2
```

```
a      1
a      2
b      1
b      2
```

kill

Se utiliza para matar procesos. Utilizar el comando **ps**, ver el **PID** del proceso a eliminar.

```
kill -9 pid_proceso
```

ln

Para poner otro nombre al fichero. Los dos nombres son equivalentes. Es como un alias. Un fichero puede tener varios nombres o enlaces. Cuando se van borrando y este número llega a 0, el fichero original se elimina.

```
ln fichero_origen alias
```

ls

Sirve la para listar los ficheros y directorios de un directorio.

Parámetros más habituales:

```
ls -a
```

Muestra los directorios: actual y padre. Representados por . y ..

ls -l

Muestra un listado en el formato largo, con información de permisos, número de enlaces asociados al archivo, usuario, grupo, tamaño, fecha y hora de la última modificación, además del nombre.

Los permisos:

drwxrwxrwx, d: Directorio, r: read, w: write, x: ejecución

Los 3 primeros para el dueño del archivo, los 3 siguientes para el grupo y los 3 últimos para otros usuarios. Los permisos se pueden cambiar con **chmod**.

lsb_release -a

Muestra la versión de Ubuntu con la que estamos trabajando.

man

Muestra ayuda sobre un comando de Linux.

man <<comando>>

man ls

mkdir

Para crear un directorio.

mkdir nuevo_directorio

nl

Número las líneas de un fichero.

nl fichero1

more

Muestra el contenido de un fichero de texto. Lo hace forma paginada, pulsando una tecla, va mostrando el resto del fichero.

more fichero_texto

mv

Sirve para renombrar un fichero.

mv nombre_origen nombre_destino

passwd

Cambiar la password del usuario actual.

paste

Para juntar ficheros, va tomando un campo de cada fichero, y los coloca en la misma línea en el fichero de salida. El comando **join** los combina, pero tienen que estar ordenados.

paste f1 f2 f3

Si f1 tiene una columna y f2 otra columna. Iría colocando en el fichero destino o salida estándar una columna de cada en cada fila. Se puede indicar un delimitador de campos:

paste -d= f1 f2 En este caso sería el **=**

pr

Prepara el fichero para imprimir. Por defecto, son 66 líneas y le formatea las cabeceras.

pr fichero1 fichero2 ... ficheroN

ps

Muestra una lista de los procesos activos. Incluye el comando ps, bash (el interprete de comandos asociado al terminal, y el resto de procesos activos). Muestra el pid de cada proceso. **ps**

Para mostrar la memoria ocupada por cada proceso:

ps aux --sort pmem

pwd

Imprime el directorio actual.

rm

Quita un nombre al fichero. En el caso de que el fichero tenga varios enlaces. Los otros nombres se mantienen. Se pueden utilizar comodines: *

rm fichero

rmdir

Borra un directorio. Si tiene ficheros u otros directorios no se puede borrar.

rmdir directorio

sort

Ordena el contenido de un fichero.

sort fichero1

split

Parte el contenido de un fichero, en trozos (de líneas). Por defecto, 1000 líneas.

Parte el fichero en ficheros de 50 líneas. Los ficheros (trozos) se nombran f.aa, f.ab ...

split -50 fichero f.

tail

Muestra las últimas líneas de un fichero de texto. Como **head** pero por el final del fichero. Mostrar las 4 últimas filas del fichero:

tail -4 fichero_texto

tr

Cambia caracteres de un fichero.

tr a x

Cambia la a por x del fichero1. Ver **tr --help**

cat fichero1 | tr a x

ulimit -s

Muestra el tamaño asignado a la Pila.

wc

Cuenta las letras, palabras y líneas de un fichero.

wc fichero_de_texto

whoami

Quién soy. El usuario con el que estoy conectado. Abreviatura de: Who am i?