

Comandos básicos linux y consejos para empezar

01

Comandos Linux básicos que usarás con frecuencia

Es sencillo recordar los comandos puesto que son las abreviaturas de ciertas palabras en inglés. Por ejemplo:

Comando linux	palabra origen	acción
ls	list	listado de ficheros
cd	change directory	cambiar de directorio
mkdir	make directory	crear un directorio
cp	copy	copiar ficheros
rm	remove	borra ficheros
rmdir	remove directory	borra un directorio
mv	mover	mover ficheros
pwd	print working directory	muestra el directorio actual
clear	limpiar	limpia el terminal
whoami	Who I am?	muestra el usuario actual

A continuación indicamos algunos comandos básicos que podéis utilizar desde el terminal. El símbolo "#" representa un comentario. El símbolo ">" es el PROMPT del sistema:

```
# mostramos la ruta inicial de directorios de nuestro usuario
# la variable HOME hace referencia a esa ruta
# $HOME indica el contenido de la variable HOME
> echo $HOME

# Supongamos que en "/$HOME/Downloads" tenemos guardada
# la carpeta "materiales", que a su vez contiene varios ficheros
# la ruta de la carpeta es, por lo tanto: "/$HOME/Downloads/materiales"

# Creamos el directorio con nombre P0-practical
> mkdir P0-practical
# copiamos la carpeta "materiales" y todo su contenido en P0-practical
> cp -rf /$HOME/Downloads/materiales P0-practical
# Nos situamos dentro del directorio P0-practical que acabamos de crear
> cd P0-practical
# Mostramos el contenido del directorio actual, incluidos los ficheros
# ocultos
> ls -la
# Mostramos el contenido de la carpeta "materiales"
> ls -la materiales
# Nos vamos al directorio "padre" de P0-practical, el que está justo "arriba"
> cd ..
```

```
# Borramos el directorio P0-practical y todo lo que haya dentro
> rm -rf P0-practical
# Creamos el directorio P1-practical
> mkdir P1-practical
# "Bajamos" al directorio P1-practical que acabamos de crear
> cd P1-practical
# movemos la carpeta de materiales al directorio P1-practical
> mv /$HOME/Downloads/materiales .
# volvemos a nuestro HOME
> cd
```



Con las **flechas** arriba y abajo del teclado, se van mostrando en el terminal los comandos anteriores o posteriores que hayamos tecleado.

Si comenzamos a escribir el nombre de cualquier fichero y pulsamos el **tabulador**, el sistema intenta autocompletar dicho nombre, por lo que no será necesario escribir tanto.

Si el nombre tiene espacios, hay que poner \ delante de cada espacio. P.ej. para mostrar el contenido del fichero "Mi fichero": > `cat mi\ fichero`

02

Tu rutina de trabajo

Durante las sesiones de laboratorio (bien en el aula o en tu casa), SIEMPRE realizarás las siguientes tareas (te aseguro que "infinitad" de veces):

1. **Editar tu programa** (por ejemplo: `fichero.cc`)

Puedes usar cualquier editor de textos que consideres oportuno: desde un editor simple, hasta un IDE con multitud de opciones. Yo no te aconsejo el IDE, puesto que en realidad solo necesitarás usar una parte muy reducida de todo lo que te ofrecerá el IDE, y puede ocasionarte muchos más problemas de los que te resuelva.

2. **Compilar** el programa que has escrito: > `g++ fichero.cc -o fichero`

3. **Ejecutar** el programa > `g++ fichero.cc -o fichero`

Dado que estos tres pasos los vas a repetir siempre, una buena forma de organizarse es usar un terminal sólo para compilar/ejecutar, y abrir el editor en una segunda ventana:

```
chdor@chdor@desktop:~$ echo Hello World
Hello World
chdor@chdor@desktop:~$
> g++ fichero.cc -o fichero
> ./fichero
```

2 y 3 Aquí SÓLO compilamos y ejecutamos

```
.bashrc (/home/hispalis70) - gedit
Archivo Editar Ver Buscar Herramientas Documentos Ayuda
Abrir Guardar Deshacer
# Alias definitions.
# You may want to put all your additions into a separate file like
# ~/.bash_aliases, instead of adding them here directly.
# See /usr/share/doc/bash-doc/examples in the bash-doc package.

if [ -f ~/.bash_aliases ]; then
fi
# enable programmable completion features (you don't need to enable
# this, if it's already enabled in /etc/bash.bashrc and /etc/profile
# sources /etc/bash.bashrc).
if [ -f /etc/bash_completion ] && ! shopt -oq posix; then
. /etc/bash_completion
fi
cat ~/mensaje
```

1 Aquí EDITAMOS el programa

Con las **FLECHAS** "arriba", "abajo" recuperamos fácilmente cualquier comando que hayamos tecleado