

Básicos

Ejemplo sintaxis de un comando

\$ prog [-aDd] [-f | -g] [-n num] req1 req2 [opt1 [opt2]]

[x] → Opcional (x puede no aparecer)

[-f | -g] → Exclusivos (a o b pero no ambos)

-f → Opción sin argumento

-n num → Opción con argumento (-n le sigue num)

req1 → Parámetro obligatorio

[opt1 [opt2]] → opt2 requiere opt1 pero son opcionales

Paths

/home/user/a/b	Absolute path
a/b	Relative path
~	Home dir
..	Parent dir
.	Working dir
-	Previous dir

Atajos de teclado

ctrl+C	→ Mata el proceso activo en shell
ctrl+Z	→ Mata (brutalmente) el proceso (si no ctrl+c)
ctrl+D	→ EOF. A la práctica cierra el shell activo
ctrl+shift+C	→ Copia texto seleccionado
ctrl+shift+V	→ Pega texto
ctrl+shift+N	→ Obre nuevo terminal
ctrl+shift+T	→ Abre terminal en nueva pestaña
ctrl+R	→ busca el comando más reciente que empieza por...

Paginas manual

Obtener ayuda sobre comandos. Mucho comandos hacen -h

\$ man [-k] program

Muestra paginas manual para <program>

-k → Busca pagina con un nombre similar

\$ whatis program

Muestra una frase de descripción

Trabajando con directorios

Operaciones de navegación y modificación

\$ pwd

Imprime el path del directorio actual

\$ cd [path]

Cambia el directorio a <path> o a *home* si no argumentos

\$ mkdir [-p] name

Crea UN directorio

-p → Crea jerarquía si hace falta. newdir1/newdir2/

\$ rmdir [-p] name

Elimina UN directorio vacío

-p → Elimina más de un directorio vacío si hace falta

\$ ln -s source destination

Crea un enlace simbólico del fichero o carpeta <source> a <destination>

Trabajando con ficheros

Creación y organización de ficheros

\$ ls [-l [-h]] [-aAtS] [path]

Lista el contenido de la carpeta actual o <path> si presente

-l → Lista con detalles

-h → Muestra los tamaños para humanos (kB, Mb, Gb...)

-a → Muestra todo, incluso ficheros ocultos

-A → Muestra todo menos los directorios [. ..]

-t → Ordena por tiempo (más recientes primero)

-S → Ordena por tamaño (descendiente)

\$ file name

Muestra el tipo de fichero (le da igual la extensión)

\$ touch name

Crea un fichero vacío con nombre <name>

\$ rm [-irf] path

Elimina permanentemente el fichero <path>

-r → Recursivamente. Elimina carpetas enteras

-i → Pide confirmación antes de eliminar

-f → Force. Ante la duda elimina sin preguntar.

\$ cp [-rif] source [sourceB...] destination

Copia un fichero de <source> a <destination>
Si más de dos argumentos se copian múltiples archivos
-rf → Ver rm
-i → Interactive. Pregunta antes de reescribir

\$ mv [-fi] source destination

Mueve un fichero de <source> a <destination>
recursivamente. Sirve para cambiar nombres
-fi → Ver cp

Trabajando con contenido de ficheros

Manipulación y filtrado de contenido

\$ cat file [fileN...]

Imprime el contenido del fichero (o ficheros)

\$ tac file [fileN...]

Como **cat** pero en orden inverso a nivel de línea

\$ head [-n] file

Imprime la 10 primeras líneas de <file>
-n → Imprime las n primeras filas (pex -3)

\$ tail [-n] file

Imprime las 10 (o n si presente) últimas líneas

\$ less file

Muestra solo el contenido de file que cabe en la pantalla y permite moverse por el sin generar scroll

\$ strings file

Imprime el contenido de <file> que es texto plano

\$ nano [file]

Editor de texto en terminal. Modifica el fichero <file> o crea uno nuevo. Ayuda en la parte inferior de la pantalla.

Miscelaneous

Manipulación y filtrado de contenido

\$ echo [-e] text [moreText...]

Imprime por pantalla los argumentos
-e → Interpreta los caracteres con \ (\n → newline)

\$ alias [name[="value"]]

Si se llama sin argumentos lista todos los alias
Si se llama con el nombre de un alias, se muestra
Si junto con el nombre se asigna un valor, se crea uno

\$ unalias aliasName

Elimina un alias

\$ passwd [username]

Cambia la contraseña (interactivo) del usuario actual o el del nombre introducido

\$ watch [-n seconds] command

Ejecuta <command> cada 2 segundos o <seconds> si existe. <command> mejor entre comillas

\$ history

Imprime el historial de comandos ejecutados

\$ clear

Limpia el texto del terminal (== ctr-L)

Processos

Listado y modificación de procesos

\$ top [-u username]

Muestra listado dinámico de procesos. Se puede filtrar por usuario.

\$ ps [-u username]

Lista los procesos iniciados desde la consola.
Si se especifica username se muestran TODOS los procesos de ese usuario

\$ kill [-9] pid

Pide al proceso con PID <pid> que se suicide. (-9 lo mata sin piedad)



\$ killall [-9] name

Equivalente a *kill* que mata a todo los procesos con Se ejecutan ambos comandos secuencialmente nombre <name>

xkill

Tras ejecutar el comando mata al proceso responsable de la ventana que se selecciona con el cursor.

Shell expansion

Operadores de control, pipes, redirección de i/o

\$ echo a ; echo b

Se ejecutan ambos comandos secuencialmente

\$ sleep 10 &

No espera a terminar la ejecución (↑ espera 10 segundos)

\$ echo \$?

Imprime el exit code del programa anterior

\$ echo a && echo b

El segundo programa solo se ejecuta si el primero ha finalizado sin error.

\$ echo a || echo b

El segundo solo se ejecuta si el primero falla

\$ echo a # echo b

Todo lo que sigue a # es un comentario, no se ejecuta

\$ echo \; \# \' \' \' \' \? *

\ es la secuencia de escape

\$ echo \$(echo "abc")

Embed shell. (↑ imprime abc)

\$!!

Ejecuta el comando anterior

\$!man

Ejecuta el ultimo comando que empiece por (man en este caso)

\$ *

Genera nombre de archivos que coinciden con cualquier combinacion de caracteres

Archivos y directorios

Archivos y directorios útiles

/etc

Archivos de configuración

/home/username

Directorio del usuario

/media

Ubicación de los dispositivos extraíble (usb, cdrom...)

/tmp

Archivos temporales

/dev/null

El disco duro más grande del mundo (solo escritura)

~/.bashrc

Se ejecuta cada vez que se abre el terminal

Pipe redirection

Redirección de salida, pipes y comandos relacionados

\$ echo hello [[> | >> | 2> | 2>\$1] fileName]

En lugar de imprimir por pantalla guarda la salida en un fichero (sobrescribiendo el contenido)

> → Redirige salida estándar, 2> → la salida de error y 2>\$1 → ambas.

>> → Redirige salida sin sobrescribir (añade al final)

\$ echo [< fileName]

< → En lugar de la entrada de teclado coge el contenido del fichero.

\$ echo a | tee filename

Guarda la entrada en un fichero Y la redirige a la salida

\$ wc [-l | -w | -c] file

Imprime la cantidad de Lines Works Characters (se puede escoger solo uno de ellos con los flags)

\$ grep [-A1 | -B1 | -v] text [file]

Imprime las líneas de <file> que contienen <text>
 -A1 → Imprime también la línea anterior (o -An)
 -B1 → Imprime también la línea posterior (o -Bn)
 -v → Imprime las que NO contengan <text>

\$ sort [-r] fileName

Imprime el contenido de <fileName> con las líneas ordenadas alfabéticamente..
 -r → Ordena en orden inverso

Network

Network

\$ curl fileURL

Imprime el contenido de una web (curl linuxupc.upc.edu)

\$ wget fileURL

Descarga un archivo de Internet

\$ ifconfig

Imprime información sobre interfaces de red

\$ ssh [username@]hostname [-p port] [command] [-X]

Conecta a un terminal remoto
 Si no se especifica username se utiliza el del usuario
 Si no se especifica puerto se coge el 22
 Si se especifica <comand> se ejecuta y se cierra sesión, sino sesión interactiva.
 -X → Permite la redirección de X11 (remote window)

\$ scp pathA pathB

Equivalente a **cp** pero uno o ambos paths puede ser remoto. En ese caso el path se especifica como: username@hostname:path

Permisos

Permisos y propiedad

\$ sudo command

Ejecuta <command> como administrador. Con “sudo su”, todo lo que se ejecuta después es como admin

\$ chgrp [-R] grpName file(s)

<file(s)> cambian el grupo a <grpName> (R → recursivo)

\$ chown [-R] ownName file(s)

<file(s)> cambian el propietario a <grpName> (R → recursivo)

\$ chmod op file(s)

Op puede ser:

nnn → Especifica todos los permisos (chmod 774)
 [u|g|o|a][+|=][rwx] → Especifica permisos individualmente. (User Group Other All, Read Write Execute)

Usuario tiene permisos rw y otros pierden el de w
 chmod u=rw, o-w
 chmod +x file # Todos tienen permiso de ejecución

Número	Binario	Lectura (r)	Escritura (w)	Ejecución (x)
0	000	✗	✗	✗
1	001	✗	✗	✓
2	010	✗	✓	✗
3	011	✗	✓	✓
4	100	✓	✗	✗
5	101	✓	✗	✓
6	110	✓	✓	✗
7	111	✓	✓	✓

Taller introductorio terminal de Linux 2017

C/Jordi Girona, 1-3 Campus Nord UPC, Edificio Omega, S-105, 08034 - Barcelona • 934137656

linuxupc.upc.edu - linuxupc@linuxupc.edu

ieee.upc.edu - ieee@ieee.upc.edu