

Manual de supervivencia en Unix/Linux/Cygwin

26th March 2004

1 Directorios

El directorio raíz se representa por `/`

El directorio home se representa por `~`

El directorio donde estoy trabajando (current directory) por `.`

El directorio de arriba de donde estoy se representa por `..`

El comando para cambiar de directorios es `cd`

`cd ~` va al *home* de la cuenta. En cygwin de prácticas es `/home/calnum`

Los directorios se pueden denominar por su camino absoluto `/home/calnum/pepe/p1`

Si estoy en `/home/calnum` por ejemplo puedo designar los directorios por su nombre relativo, por ejemplo `ls ./pepe/p1`

`cd ../../p2` sube dos veces y baja a `p2`

2 Comandos

`pwd` Escribe directorio donde estoy

cat fich Lista fichero *fich* en pantalla
mkdir midir Crea directorio *midir*
rmdir midir Borra directorio *midir* si está vacío
ls midir Lista ficheros en el directorio *midir*
ls -l midir Lista dando detalles
ls -a midir Lista ficheros que empiezan por .
mv dir1/fich1 dir2/ Mueve(sin copiar) *fich1* de *dir1* a *dir2*
mv dir1/fich1 dir2/fic2 Mueve *fich1* de *dir1* a *dir2* y le cambia de nombre a *fich2*
cp dir1/fich1 dir2 Copia *fich1* a *dir2*
cp -r dir1 dir2 Copia *dir1* de forma recursiva (con todos los directorios y ficheros a *dir2*)
cp fich A: copia fichero *fich* en diskette A: (solo Cygwin)
rm fich Borra fichero *fich*
rm -r dir Borra directorio *dir* de forma recursiva(sirve para borrar directorios no vacíos con todos los ficheros y directorios que contengan).
less hace que un comando imprima solo una pantalla, y se espere a que se haga ENTER para imprimir la siguiente. Es útil en listados muy largos. se utiliza de la forma
cat fich|less
ls dir|less
 Para ficheros muy largos es útil el saber el principio y el final sin editarlos. Esto se consigue con
tail fich Escribe las últimas líneas en pantalla.
head fich Escribe las primeras líneas en pantalla
 La salida de un ejecutable se puede redireccionar a un fichero en vez de la pantalla. Si queremos que el ejecutable *prog.exe* realice la salida en *sal.txt* hacemos
prog.exe>sal.txt

La desventaja es que hay que entrar los datos a ciegas pues no se ven en pantalla las salidas del programa pidiendonos datos. Esto se evita poniendo los datos en un fichero y redireccionando la entrada al fichero con <. Si ponemos los datos en *input.dat* ejecutaríamos el programa como

prog.exe<input.dat>sal.txt

tar -zxvf fich.tgz Extrae los ficheros de un archivo comprimido .tgz

./myprog& Ejecuta myprog en el background(sin bloquear la pantalla)

g++ myprog.cpp -o myprog compila myprog.cpp creando el ejecutable myprog

make Ejecuta los comandos en el fichero *Makefile*