
PROCESSES

^c <ctrl>-c kills (definitely stops) current job
^z <ctrl>-z suspends the current job. This can either be moved to the background or resumed in the foreground by using **bg** or **fg**

bg moves the current process to the background
fg moves a process to the foreground. (If there is more than one suspended job, use **jobs** to decide which you want to **fg**)

fg 2 moves process number 2, as listed by **jobs**, to the foreground

jobs lists background and suspended processes (created with **bg** or **^z**)
jobs -l ("el" not one) includes the pid (process id number)

ps lists all your processes

kill stops a process (use **ps** or **jobs** to find your processes)
kill 2986
kills off the process with pid 2986

MISCELLANEOUS

finger tells you who is logged on (see also **w**)

w shows information about logged in users

who produces similar result (see **finger**)

tar create (or extract) a tarball from (to) a list of files
tar -cvf tarball.tar subdir/*
tar -xvf tarball.tar
the option **-z** compacts the files by **gzip**

wc word count
wc long.file
prints the number of lines, words and characters in *long.file*. Options include **-l** to count lines only, and **-c** to count characters only

ln create a link or an alias for a file
ln -s subdir/orig.file alias.file

history displays last several commands used
!! re-executes the last command
!51 executes command 51 in the history list use also **<up>** - and **<down>** - arrows to navigate in the history

date displays current date and time

passwd invokes a password changing program

exit leaves the current shell (same as **^d** or **<ctrl>-d**) usually = logout

GRAPHIC DISPLAY

To display graphics, most Unix require the configuration of the X-Window server.

Commands on your local computer:

xhost set the list of allowed X-Window clients
xhost +
The "+" allows any remote computer to display on your local display

ifconfig gives information about the network configuration (e.g., the current IP_address, usually similar to 123.145.167.189)

Commands on the remote computer:

setenv set up an environment variable (tc-shell)
setenv DISPLAY IP_address:0.0
required to tell the remote computer where it should display its graphics

xclock starts a graphic clock (e.g., used to test the X-Window server or to get the current time... ;-)

This document was originally written and designed by Aoife McLysaght and Andrew Lloyd© from the Irish EMBnet node, and modified by Laurent Falquet from the Swiss EMBnet node and distributed by the Publications Committee of EMBnet.

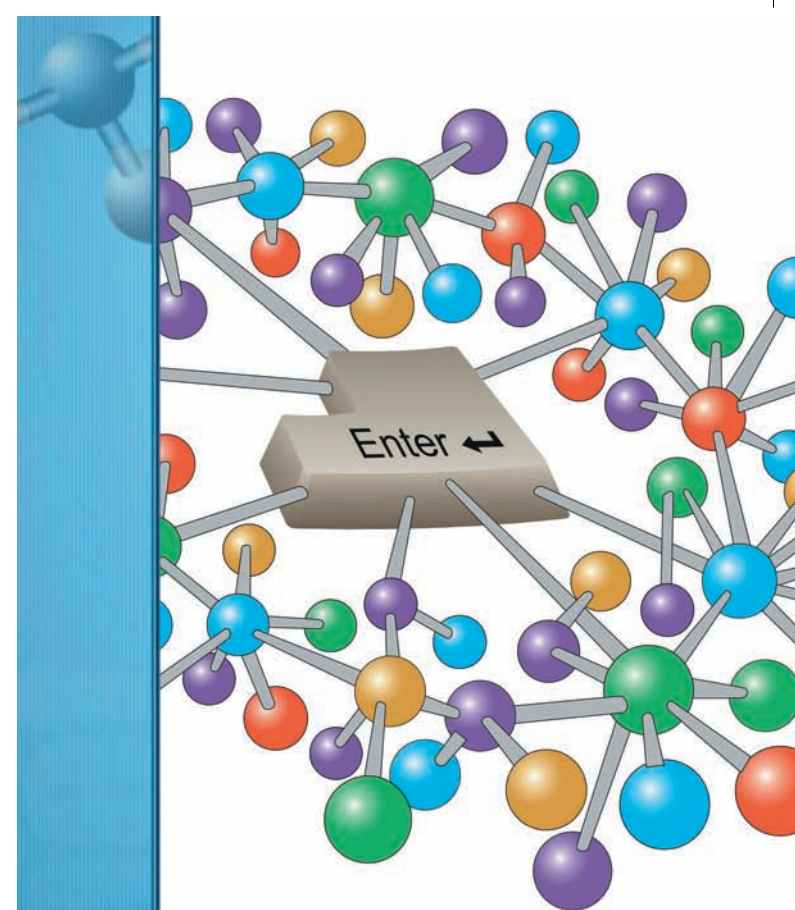
EMBnet - European Molecular Biology network - is a network of bioinformatics support centres situated primarily in Europe. Most countries have a national node which can provide training courses and other forms of help for users of bioinformatics software.

Further information about UNIX is available from your national node. You can find contact information about your national node from the EMBnet web site:

<http://www.embnet.org/>

If you have found this publication useful, please let us know.
If you have ideas for similar documents we'd like to hear from you: emb-pr@embnet.org

A Quick Guide To UNIX
Revised edition 2003



A Quick Guide UNIX

EMBnet

A Quick Guide To UNIX

This is an introduction to the UNIX operating system. Unix may seem idiosyncratic, even impenetrable, to begin with but it has the virtue of minimising the number of keystrokes and so speeding up your access to the computer.

The commands listed here are common to different operating systems and shells. They include some of the most useful and frequently used commands in UNIX. The power and utility of most UNIX commands can be enhanced with switches or options preceded by a “-” sign.

More information on the options, the effects and how to use the commands is available by using the **man** command:

man gives manual information on a topic
man grep
displays the manual page about grep
apropos lists all the man(ual) entries relating to a topic (same as **man -k**)
apropos print

Another useful source of information is the on-line EMBnet tutorial which includes a page on UNIX

<http://www.dk.embnet.org/Embnetut/Univsl/unixcmds.html>
or equally

<http://www.uk.embnet.org/Embnetut/Univsl/unixcmds.html>

The general format of this document is that anything in **bold** is a command you can enter. Anything in *italic* is a fake file or directory name you must change according to yours. Anything preceded by a hyphen “-” is an option which will modify the effects of a command. A general description of each command is followed by one or several examples of its use.

FILES

ls lists files in a directory
ls -alF
lists **-a** all files in **-l** long format **-F** identifies directories **/**, executable files ***** and symbolic links **@**, in the current directory
cat concatenates and displays files
cat my.file
displays *my.file* on the screen

chmod modifies the read (**r**), write and delete (**w**), and execute (**x**) permissions of specified files and the search permissions of specified directories. The permission can be set for user (**u**), group (**g**) or other (**o**)
chmod go-w my.file
stops (**-**) anyone else (**go**) changing or deleting (**w**) *my.file*
chmod g+rx my.file
allows (**+**) anyone of my group (**g**) reading, changing, deleting or executing (**rx**) *my.file*
cp copies files
cp orig.file copy.file
cp orig.file subdir/new.file
copies *orig.file* to *new.file* in *subdir* directory
cp subdir/orig.file .
copies *orig.file* from *subdir* to the current directory (**.**) without changing its name
mv moves/renames a file (or directory)
mv oldname newname
mv my.file subdir/my.file
a move (**mv**) is equivalent to a copy (**cp**) followed by a remove (**rm**)
rm removes/deletes a file.
rm oldfile
rm -i *.file
option **-i** (interactive) advised if wildcards (*****) in use
diff compares two files and prints how they differ
diff file1 file2
prints differences to screen options include **-b** to ignore differences in blank space, and **-i** to ignore case
find searches the directory tree for a file
find . -name lostfile -print
will search your current directory (**.**) (and any subdirectories) for *lostfile*
grep searches a file for a string
grep word my.file
grep "two words" my.file
options include **-i** to ignore case and **-n** to print line numbers
vi simple screen oriented text editor

pico simple display oriented text editor
pico myfile.txt
head prints the first few (default = 10) lines of a file
head oddfile
head -20 oddfile
displays first twenty lines of *oddfile*
tail displays last few lines of a file (see head)
more displays a file one screenful at a time
more longfile
hit **<spacebar>** to see the next screen
Note: some people prefer **less**

OUTPUT REDIRECTION

> redirects output of a command to a file
diff file1 file2 > new.file
puts differences into *new.file*
cat one.file two.file > both.file
writes the output of the cat command into *both.file* (overwrites *both.file*)
>> appends a file to the bottom of another
cat three.file >> both.file
appends *three.file* to the bottom of *both.file*
| “pipe” - uses the output of the first command as the input of the second
grep string my.file | wc -l
finds how many lines on which “*string*” occurs (see **grep** and **wc**)

DIRECTORIES

cd changes current directory
cd /etc
go to */etc* directory
cd ..
go up one level in directory tree
cd ../subdir2
go “sideways” to *subdir2*
mkdir creates a new subdirectory
mkdir subdir
rmdir removes a directory - you must delete all the files in it first
rmdir subdir
pwd print working directory, tells your current location (path)