

UNIX BASICS


It is possible to get a basic understanding of UNIX in about 20 mins. Print out this little text, get yourself access to a UNIX machine (usually sorted out by a computer administrator), make yourself a cup of coffee and try out these instructions. If a command doesn't work, don't worry: just try the next one (i.e. they're not sequential).

Open a terminal window* and try typing any of the following commands at the command prompt (the "prompt" is just some text which appears on the screen to indicate you can type in commands there, usually ending in "\$", ">" or "%"). Also, remember that

- In a terminal, up- and down-arrows can be used to get previously-typed commands (the "history" feature - mostly used to modify and reuse a previous command).
- If you press TAB while typing a directory or file name, the system will try to complete it for you ("tab completion", http://en.wikipedia.org/wiki/Tab_completion).
- UNIX is always case-sensitive (so "File1" is not the same as "file1") and file names with spaces go in quotes (e.g. emacs "My File.txt").
- Always use the full file name (e.g. "acread Paper1.pdf", not "acread Paper1").
- If UNIX seems to have crashed, try CTRL+z or CTRL+c to break out of it (*use with care!*)

by Toby Mathews
August 2008

* Depending on your system, terminal windows are opened different ways:

- If you are using UNIX on your own computer (e.g. LINUX) then just open a fresh terminal (usually an icon like  "terminal" on LINUX). If you are using a remote link to a UNIX workstation/server, open a terminal and type (usually) "ssh UserLogin@HostIP" where HostIP is the IP address of the machine you want to use (e.g. I've used pisa.ouce.ox.ac.uk, sysa.abdn.ac.uk, idefix1.saclay.cea.fr) and UserLogin is your username on HostIP.
- If you use Windows, it's best to download and install a program to handle your log-in: I've variously used PuTTY (see link below) or SSH (<http://www.gsw.edu/~oit/techsupp/ssh.html>) with Xming (<http://www.straightrunning.com/XmingNotes>), or Exceed (<http://connectivity.opentext.com/products/exceed.aspx>). I also advise you to install a file transfer program, e.g. FileZilla (<http://filezilla-project.org/download.php?type=client>) or WinSCP (<http://winscp.net/eng/download.php>) or FreeCommander (<http://www.freecommander.com/>).
- If you are using Windows at the OUCE (<http://www.ouce.ox.ac.uk>), then this is the recommended way to connect (formerly on <http://itfaq.ouce.ox.ac.uk/internal/db.php?No=663#FAQ>):
 1. Start *Exceed* running: click on the "Hummingbird Connectivity" shortcut, which should be on the Desktop, go into the Exceed directory and double-click on "Exceed" (not "Exceed XDMCP Broadcast" or "Exceed XDMCP Query"). Missing out this step means that you cannot use programs that try to open a separate graphics window (i.e. you can still use all shell commands below).
 2. If you do not have PuTTY on the Desktop, download it from <http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html> (I use "Windows installer for everything except PuTTYtel").
 3. Start up PuTTY and enter HostIP in the Host Name field. In the Category list, select Connection and choose to send keepalives every 120 seconds (see <http://the.earth.li/~sgtatham/putty/0.58/htmldoc/Chapter4.html#config-keepalive>), select Connection->SSH->X11 and ensure that "Enable X11 Forwarding" is checked (you can save these settings using the "Save" button in the Session category).
 4. Click on the Open button and a terminal window will appear asking you for your UserLogin and your password. Once past those, you should have a command line prompt of the form "[UserName@Host CurrentDir]\$".
 5. Now you're logged in and you can enter UNIX commands. If it's your first time, I suggest you try some of the commands in the table below to see what they do. If you're comfortable with UNIX commands already, you can start up some of the installed software by entering its command (e.g. "idle &", "matlab &").

UNIX Command	What it should do
ls	List the contents of the current directory.
ls -l	Same as ls, but with more detail (for an explanation of the "rwx" stuff this command produces, see http://networking.ringofsatum.com/Unix/unixpermissions.php ; "ls -al" to list hidden files too).
cd dir1	Move into directory "dir1" contained in the current directory.
cd dir1/dir2	Move straight into directory dir2 contained in dir1.
cd ..	Move out of the current directory.
pwd	Tells you what directory you're currently in (your current "path").
passwd	Change your login password (<i>use with care!</i>).
clear	Clear the screen.
date	Print date and time.
cd /	Move to the "root" of the directory tree.
cd ~	(or just "cd") Move to your "home directory" (where you started when you logged on).
man XXX	Get help on command XXX (e.g. put "ls" instead of XXX; move through the help page using ENTER, space bar or q). "man" gives you more information on the commands listed here and extra features that I haven't told you about.
apropos YYY	List all commands anything to do with YYY (e.g. put "quota" instead of "YYY").
cp file1 file2	Copies file1 and saves it as file2 (you can use "?" or "*" as wildcards in filenames).
find dir1 -iname "toby*"	Find all files in dir1 whose filename starts with "toby"
mv -i file1 file2	Renames file "file1" as "file2".
rm -i file1	Deletes file1.
mkdir dir3	Creates a directory called "dir3".
rmdir dir3	Deletes dir3 (if dir3 is not empty, you'll need to use "rm -r dir3" instead).
cat > file3	Creates a textfile called "file3" and puts into it whatever you type in (when you're finished, go to a new line and press CTRL+d).
more file3	Display textfile file3 (try "less" if "more" doesn't work). Use "head file3" or "tail file3" to see just the start/end of file3 (useful for large files).
cat file3 >> file4	Append file3 to file4.
lp file3	Print file3 (check on progress with "lpstat -p all").
history	List the commands you have recently entered ("history -c" to clear the record).
grep -i "XXX" file3	Lists all the lines of file3 that contain the search string "XXX".
grep -ir "XXX" *	Search all files in the current directory and sub-directories for all occurrences

UNIX Command	What it should do
	of “XXX”. Remove the “i” for a case-sensitive search.
pine	A text-based email program (if “pine” doesn’t work try “ream” or “elm”; again, read the program’s web pages to learn how to use it, e.g. http://www.washington.edu/pine/).
emacs file3	Edit file3 in the Emacs text editor program (use CTRL+x CTRL+c to exit the program; also try “ue”, “pico”, “textedit”, “xemacs” or “vi” instead of “emacs”, but if the right programs haven’t been installed then some of these commands won’t work; read the program’s help/web pages to learn how to use it, e.g. http://wings.buffalo.edu/computing/Documentation/gen/ref/emacsref.html for Emacs, http://wings.buffalo.edu/computing/Documentation/gen/ref/virefcad.html for vi). Use “emacs file3 &” to run this in a separate window (allows you to continue working on something else at the same time).
prog1	To run an “executable” called prog1, either just type its name “prog1” (or perhaps “./prog1”). With “time prog1” (or “time ./prog1”) you will also get a report of how long the run took. “nohup prog1 &” means the job will not abort when you log out.
sh script1.sh	Try this (type <i>exactly</i> what I’ve put here): “cat >script1.sh” then ENTER then “date” then ENTER then “ls -l” then ENTER then CTRL+d then type the command “sh script1.sh”. This is scripting and is a very important part of UNIX (you can see the “script” you have just created using emacs). To teach yourself scripting, read http://www.scribd.com/doc/1251010/Advanced- BashScripting-Guide .
dos2unix script2.sh	Converts Windows textfiles to UNIX textfiles (this command wouldn’t affect script1.sh above, but if you have another script script2.sh which has been edited using e.g. Windows Notepad and then uploaded directly onto the server, it must be converted before sh can read it).
<i>Editing textfiles</i>	A large amount of work on UNIX involves writing and editing textfiles (always UNIX-format textfiles: never use Windows-format textfiles). If you’ve heard of people talking about scripts (.sh or .bat files), C programs (.c, .cpp) or FORTRAN programs (.f, .f90), these are all just textfiles that are edited using a text editor like emacs. C and FORTRAN programs are then “compiled” to create an executable file, which you execute as described above. This is what “programming” is.
diff file1 file2	Very useful for textfiles. If file1 and file2 are very similar (e.g. two different versions of the same program) then diff will list the lines where they differ.

UNIX Command	What it should do
acoread Paper1.pdf &	Open the file Paper1.pdf in a new window (the “&” runs acoread as a new “job” in a new window).
firefox Page1.html &	Open the file Page1.html in a new window.
zip -r file4 dir4	Create a “zip” file called “file4.zip” and store everything in directory dir4 in it (this is the same format as WinZip).
unzip file5.zip	This will ‘unzip’ a zip file “file5.zip” and put its contents in the current directory (more on file compression at http://en.wikibooks.org/wiki/Guide_to_Unix/Commands/File_Compression).
who	Lists all the users currently logged in on the system.
ps -f	Shows all processes you are currently running. Try typing “xclock &” then “ps -f” then “kill -9 X” where “X” is xclock’s process ID (PID column).
jobs	Shows all jobs you are currently running (for more about jobs and processes, see http://www.ncl.ac.uk/iss/unix/unixhelp/job_process.html).
chmod -R u=rwx dir4	Change permissions on directory dir4 and any subdirectories inside it (use “ls -l” before and after to see what changes) (replace “dir4” with a list of space- separated file names to change specific files only). Also try “u=rwx,go=rx” instead of “u=rwx” (the “u=”, “g=” and “o=” bits specify what the user (you), your group and others (i.e. anybody) can do to the files in dir4: see http://www.zzee.com/solutions/chmod-help.shtml). You can also use octal codes for short, e.g. “700” is the same as “u=rwx”, “755” as “u=rwx,go=rx”.
quota	Show your allowed disk usage(s) in Kb.
du more	Shows the size of all your directories in Kb (the last line shows your current disk usage in Kb) (“ ” is called “pipe” in UNIX: it tells the computer to take the output of the previous command and use it for the next one, in this example the “more” command).
free	Show how much memory you have on this computer (also try “grep MemTotal /proc/meminfo”)
history grep cd	Finally, two “pipelining” examples for you to ...
who wc -l	... try out: what do they do and how do they work?
logout	The end.

There are, of course, plenty more UNIX commands (see the “UNIXHelp for Users” project at <http://unixhelp.ed.ac.uk>), but I find that the ones listed here are by far the most common.

Lastly, see http://en.wikipedia.org/wiki/The_UNIX-HATERS_Handbook for a long list of UNIX’s shortcomings and see <http://www.linux.com/howtos/Secure-Programs-HOWTO/history.shtml> for a bit about where UNIX came from, if you are interested.