

Appendices

Command summary

This is a very quick summary of all the command line commands we cover in this course.

Command Example	Action
bash	bash commands.sh
bg	This is the name of the command line interpreter we've been using all along!
cat	cat abc.txt def.txt
cd	cd ../Work
chmod	chmod a+x commands.sh
clear	Background a command stopped with [Ctrl]+[Z].
cp	cp island.jpg map.jpg
date	date +"%H:%k:%M"
echo	echo *.txt
env	Concatenate one or more files.
exit	Change directory.
export	export TERM=xterm
fg	Change the mode (permissions) of a file. Use "chmod a+x" on a shell script.
grep	grep rum story.txt
history	Clear screen. (Better to use [Ctrl]+[L].)
jobs	Copy a file. (Use the "-R" option to recursively copy a directory.)
less	less lorem.txt
ls	ls Work
mkdir	mkdir Fun
more	more lorem.txt
mv	mv island.jpg ../map.jpg
rm	rm nonsense.txt
rmdir	rmdir Fun
scp	Remove an <i>empty</i> directory.
sftp	Remotely copy a file or directory tree to or from a remote system.
ssh	Interactively transfer files to or from a remote system.
touch	touch newfile.txt
type	type more
unset	unset TERM ⁹
wc	wc lorem.txt

We also met a number of graphical applications:

Command	Default for these file types	
eog	GIF, JPEG, PNG, SVG	"Eye of GNOME" graphics viewer
evince	PDF	Document viewer
firefox	HTML	Firefox web browser
gedit	TEXT	Text editor
xeyes		Silly test application to track the cursor

⁹ "unset TERM": *Don't do this!*

Date formats

Year

%C	20	Century
%Y	2009	Four digit year
%y	09	Two digit year

Month

%b	Apr	Abbreviated month name
%B	April	Full month name
%m	04	Two digit numerical month

Day

%j	118	Day of year (1...366)
%d	28	Two digit day of month
%a	Tue	Abbreviated day of week
%A	Tuesday	Full day of week
%u	2	Numerical day of week (1...7, 1=Monday)
%w	2	Numerical day of week (0...6, 0=Sunday)

Hour

%H	21	Hour of the day (0...23)
%I	09	Hour of the day (1...12)

Minute

%M	07	Minute of the hour
----	----	--------------------

Second

%S	23	Second of the minute
%s	1240949377	Seconds since 1970-01-01 00:00:00 GMT

Useful

%n		New line
%t		Tab

There are two useful modifiers. If "%M" were to give "07" then "%_M" would give "_7" and "%-M" would give "-7".

Globbering

*	Any string of characters, including the empty string.
thing*	matches: thing.txt , thing , things does not match: thin, think
th*ing	matches: thinking , thing , the_ shining does not match: ting, think
th*in*	matches: thing.txt , thing , things , thanking , the\ shining , thin , thinks
?	Any single character.
thing.???	matches: thing.txt , thing.dat , thing.jpg does not match: thing.jpeg, thing
[...]	Any single character from the set in the brackets.
thing.t[xyz]t	matches: thing.txt does not match: thing.tot
[^...]	Any single character not in the set.
thing.t[^xyz]t	matches: thing.tot does not match: thing.txt

The character class globbing expressions [:...:] all appear inside the standard [...] brackets of the glob, so we get doubly nested square brackets. So "[N[:digit:]]" matches "N" or "any one digit".

[:alnum:]	Any alphabetic character (upper or lower case) or any digit.
[:alpha:]	Any alphabetic character (upper or lower case).
[:blank:]	Any horizontal white space (space or tab, essentially).
[:digit:]	Any of the ten digits.
[:lower:]	Any lower case alphabetic character.
[:upper:]	Any upper case alphabetic character.

PS1 codes

These codes can be placed inside the PS1 environment variable. There are more, but these are the useful ones.

Time codes

<code>\D{format}</code>		The date where the format is given by a format string using the %-codes listed above.
<code>\d</code>	Tue 10 Sep	The date in this specific format
<code>\t</code>	17:28:26	24-hour time, with seconds
<code>\T</code>	05:28:26	12-hour time, with seconds
<code>\A</code>	17:28	24-hour time
<code>\@</code>	05:28	12-hour time

Other codes

<code>\h</code>	pcphxtr01	Short machine name
<code>\H</code>	pcphxtr01.pwf.cam.ac.uk	Full machine name
<code>\l</code>	2	Terminal number, /dev/pts/2→2
<code>\u</code>	y250	User name
<code>\W</code>	work	Name of the current working directory
<code>\w</code>	/home/y250/Unix Intro/Work	Absolute path of the current working directory

Command line cursor control

There are often two ways to do these operations. One way avoids the use of the cursor keys but requires the memorisation of some other letters.

<code>[Ctrl]+[F]</code>	<code>[→]</code>	Move right one character.
<code>[Ctrl]+[B]</code>	<code>[←]</code>	Move left one character.
<code>[Alt]+[F]</code>	<code>[Ctrl]+[→]</code>	Move right one word.
<code>[Alt]+[B]</code>	<code>[Ctrl]+[←]</code>	Move left one word.
<code>[Ctrl]+[A]</code>	<code>[Home]</code>	Move to start of line
<code>[Ctrl]+[E]</code>	<code>[End]</code>	Move to end of line.
<code>[Ctrl]+[W]</code>		Remove the word to the left of the cursor.
<code>[Ctrl]+[K]</code>		Remove the line to the right of the cursor.
<code>[Ctrl]+[U]</code>		Remove the line to the left of the cursor.
<code>[Ctrl]+[P]</code>	<code>[↑]</code>	Go back one line in history.
<code>[Ctrl]+[N]</code>	<code>[↓]</code>	Go forwards one line in history.

sftp commands

Any Unix command can be run on the local system by preceding it with a "!". Where there is no `l` - local version of a command we show the `!` - version. The `!` - version always works, except for "`lcd`".

Remote	Local	
<code>cd</code>	<code>lcd</code>	Change directory
<code>ls</code>	<code>lls</code>	List directory contents
<code>pwd</code>	<code>lpwd</code>	Print working directory
<code>mkdir</code>	<code>lmkdir</code>	Make a directory
<code>rmdir</code>	<code>!rmdir</code>	Remove empty directory
<code>rm</code>	<code>!rm</code>	Remove file
<code>get remote_name</code>		Fetch a remote file, keeping its name.
<code>get remote_name local_name</code>		Fetch a remote file, changing its name.
<code>put local_name</code>		Put a file onto the remote system, keeping its name.
<code>put local_name remote_name</code>		Put a file onto the remote system, changing its name.
<code>help</code>		Show the complete set of sftp commands.
<code>quit</code>		Quit sftp.

Environment variables

HOME

Specifies your home directory.

You should never change this value.

```
pcphxtr01:~$ echo "${HOME}"  
/home/y220
```

PATH

Specifies the list of directories where the operating system goes looking for executable files to run the commands you issue.

You should only ever add to this value. Removing directories from it that are provided by the system may break some system facilities. On PWF Linux and OpenSUSE Linux your `${HOME}/bin` directory is added to your `PATH` by the system if and only if it exists when you log in.

```
pcphxtr01:~$ echo "${PATH}"  
/home/y220/bin:/usr/local/bin:/usr/bin:/bin:/usr/bin/X11:/usr/X11R6/bin:/usr/g  
ames:/opt/kde3/bin:/usr/lib/mit/bin:/usr/lib/mit/sbin:/opt/novell/iprint/bin:/  
opt/real/RealPlayer
```

PS1

Specifies your shell prompt.

Characters preceded by a backslash, "\", are translated into system data. Ordinary characters are used unchanged. A trailing space is often a good idea.

\h	The machine name, also known as the host name.
\t	The time
\w	The name of the current w orking directory.
\\$	This is just the same as "\$" if you are not the super user. It changes to "#" if you are.

```
pcphxtr01:~$ echo "${PS1}"  
\h:\w\$
```

TERM

Specifies your terminal type.

This should be set by the system and you should not need to change it. Commands that need to know the parameters of your terminal will fail if this is unset or incorrectly set. (e.g. `more` needs to know how many rows your screen has.)

```
pcphxtr01:~$ echo "${TERM}"  
xterm
```