

Linux quick reference card

Conventions

Optional command arguments are enclosed in square brackets []. Multiple arguments are separated by spaces unless otherwise stated. Command options can sometimes be combined – see sect. 1 for information on how to get details for specific commands. *Emphasised* words are place holders for real filenames etc. Section symbols (§) denote references to other sections in this card.

1 Help

Online help is available via the manual commands `kdehelp`, `xman` and `man`

`kdehelp` invoke the KDE manual browser

`xman` invoke the window based manual browser

`man command` lists information on *command*

`man -k keyword` lists all commands referring to *keyword*

2 Work session

2.1 Logging in

In the CS1 Laboratory. Type your *username* in the Login: text-box and your *password* in the Password: box.

From an EUCS Lab. Ask the demonstrator for help with using the `eXceed` application.

2.2 Changing your password

The command `yppasswd` prompts for old/new passwords.

2.3 Logging out

`logout` or **CTRL+D** (§3.1, 7.4)

`login [username]` — logout and login user *username*

3 Keyboard

3.1 Control Keys

The following control keys may be used to edit the command line and to affect the behaviour of running programs.

Key	Function
CTRL+A *	Moves to start of line
CTRL+C	Interrupts many programs
CTRL+D	End-of-file character (logout)
CTRL+E *	Moves to end of line
CTRL+R *	Search for previous commands
CTRL+Z	Suspends foreground job (§7.2)
TAB *	Completes filenames, commands
↓	Displays previous command in command history (§7.3)
↑	Displays next command

* denotes a *bash* control-key sequence (§7).

4 The File System

4.1 Files

4.1.1 Filenames

Filenames may be chosen from any combination of alphanumeric characters plus “-”, “_” and “.” (other characters such as control characters or spaces may be used but need to be “escaped” using the “\” character).

In all commands, files may be referenced by their *full path names* or by a *relative path name* e.g. `/home/cs1/filename` is a full path name (starts with a forward slash) but `filename` and `../my-dir/filename` are relative path names (§7.2).

4.1.2 File management commands

`cat [-n] f1 [f2 ..]` — concatenates and lists entire files [-n with line nos.]

`more filename` — lists file a page at a time (within more, type ? for help, space bar for the next page, q to quit). (§8).

`ls` — lists files in current directory

`ls [opts] name` — lists file or directory *name*

Options: `-F` lists files appending “/” if a directory and “*” if executable, `-l` gives information on size, owner, permissions and modification time, `-R` recursively lists directory, `-a` lists all files starting with “.”.

`mv f1 f2` — renames *f1* as *f2*

`mv f1 [.. fn] d1` — moves *f1* [.. *fn*] into directory *d1* Renaming files to be the same as existing files is prohibited if the `noclobber` variable is set (§7.4).

`cp f1 f2` — copies *f1* to *f2*

`cp [-r] f1 .. fn d1` — [recursively] copies *f1* .. *fn* into directory *d1*

`rm fname` — remove *fname*
`rm -r dname` — recursively remove *dname* (§4.2)

4.2 Directories

Directories can be listed, renamed and copied just like ordinary files (§4.1.2)

`mkdir dname` — makes a new directory *dname*
`rmdir dname` — removes directory *dname* only if it is empty (to remove non-empty directories (§4.1.2))
`cd` — return to your home directory
`cd dname` — change directory to *dname*
`pwd` — echoes your current working directory

4.3 File permissions

Each file (and directory) has nine bits specifying its access permission. Access can be given for a user to *read*, *write*, or *execute* a file (or any combination of these). Permissions can be specified for the *owner* of a file, the members of a file's *group* and for everyone else. (`ls -l` provides this information). The `chmod` command is used to change these file modes.

`chmod [class(es)] op perm(s) [, . . .] file(s)` where
 classes are *u*, *g*, *o*, *a* (user(owner), group, others, all), *op*(erations) are *=*, *-*, *+* (set permission, remove access, give access) and *perm*(issions) are *r*, *w*, *x* (read, write, execute).

e.g. `chmod u=rwx,g=rw,o-rwx fl`

4.4 File quotas and disk usage

`du [dir]` displays disk usage for current [*dir*] directory
`quota` displays user's disk usage if over limits
`quota -v` displays user's disk usage

5 Printing

`lpr [-Ppname] [-d] fname` — prints file *fname*

Options:

`-Ppname` prints on printer *pname* [otherwise the default printer as specified by the `PRINTER` environment variable (§7.4)].
`-zoption-list` gives access to a whole range of printing options to allow double sided output (on some printers) and so on.

`lpq` — lists the default printer queue

`lpq -Ppname` — lists the queue for printer *pname*

`lprm -Ppname job` — removes job number *job* from the queue for printer *pname*

`a2ps -Ppname [-2r] [file]` — converts standard input [or *file*] into postscript for printing on laserprinters. [-*2r*] option puts 2 pages on a single A4 sheet. `-P` options as for `lpr`

The printer in the CS1 Laboratory is named `ljet` (Laser Jet at the Appleton Tower).

6 Processes

`ps -x` list all processes belonging to you
`kill -9 pid` kill process number *pid* (*pid* obtained from `ps -x`)
`w` list all users on the machine
`whoami` owner of this shell

7 bash

7.1 Commands

`alias astring = cstring` abbreviate command *cstring* to be *astring*
`bg [n]` place last [*nth*] suspended job in background
`exec command` replace shell with *command*
`fg [n]` bring last [*nth*] suspended job into foreground
`history` list command history (§7.3)
`login, logout` (§2)
`var = value` set variable *var* [to value *val*]
`export var` export value assignment globally (§7.4)
`source fname` run commands in *fname*
`umask ddd` sets file permission creation mask (§4.3).

7.2 Special Characters

7.2.1 Filename substitution

You can abbreviate filenames using *wild card* characters.

`?` matches any single character
`*` matches an arbitrary number (≥ 0) of characters
`[. . .]` matches any single character
 e.g. `[an]` matches a or n
`[a-z]` matches any lowercase letter

7.2.2 Filename abbreviations

`.` current working directory
`..` parent of current working directory
`~` your home directory
`~user` home directory of *user*

7.2.3 Redirecting Input and Output

`cmd > fname` redirects output from `cmd` into `fname` (replacing `fname` if it exists)

`cmd >> fname` redirects output from `cmd` and appends it to `fname` (created if it doesn't exist)

`cmd < fname` uses `fname` as input for `cmd`

With both the `>` and `>>` commands above, if `noclobber` is set (§7.4), use `>!` or `>>!` to force overwriting of the file.

7.2.4 Pipelines

The output stream from one command may be piped into the input stream for another using the “|” symbol. For example `ls | wc` counts the files in the current directory.

7.2.5 Jobs

`jobs` lists all background jobs

`command &` runs `command` in background

`CTRL+Z` stop the foreground job

`bg` places the stopped job in the background

`% [n]` brings the current `[nth]` background job to foreground

Also (§7.1).

7.3 History mechanism

Special characters:

`!!` — entire last line

`!n` — command line `n` in history

`!$` — final word on last line, e.g. `ls !$`

`!*` — all of last line except for first word

`!?pat?` — most recent line containing `pat`

All of the above can be modified with the following suffixes:

`:n-m` — words `n` to `m` on line e.g. `!!:2-3`

`:p` — list command without executing, e.g. `!foo?p`

7.4 bash variables

Modified using `var = value` (§7.1). The string `$var` returns the value contained by variable `var`. The following tables list some of the more commonly used variables.

Variable	Use
<code>history</code>	number of previous commands retained by history mechanism (§7.3)
<code>ignoreeof</code>	if set, prevents shell from being killed by a <code>CTRL+D</code>
<code>noclobber</code>	if set, prevents files from being overwritten by <code>></code> and <code>>></code>
<code>savehist</code>	contains number of commands to save in <code>.history</code> file on shell exit
<code>EDITOR</code>	name of your editor
<code>PATH</code>	contains the list of directories to search for issued commands
<code>PRINTER</code>	contains name of default printer
<code>user</code>	userid of current process
<code>HOME</code>	your home directory (§7.4)
<code>TERM</code>	contains terminal information

Note the list below the line should be regarded as *read-only* variables.

8 Miscellaneous filters

Most of the following commands may be used as *filters* i.e. they can take their input from the *standard input* (and thus be part of a pipe (§7.2.4)). See the manual pages for details.

`grep pattern file(s)` — types out the lines containing `pattern` in the `file(s)`. (`egrep` is a faster, similar command allowing extended regular expression matching).

`grep -v pattern file(s)` — types out the lines **not** containing `pattern` in the `file(s)`

`head [n] file(s)` — types out the top `n` lines of `file(s)`

`tail [n] file(s)` — types out the last `n` lines of `file(s)`

`diff file1 file2` — lists the differences between the two files. One of the filenames may be replaced by a hyphen to specify standard input.

`sort [-n] [-r] file` — sort `file` in `[numeric]` ascending `[descending]` order.

`wc file(s)` — counts characters, words and lines in text files

9 Generic Startup Files

These files are used to setup the environments of certain programs.

.history Shell history saved from last session

.newsrc A record of the news items you have read.

10 K Desktop Environment

The K Desktop Environment allows you to manage several Linux applications running at once on your

machine. It has a taskbar at the top of the screen and the *kpanel* at the bottom, providing access to applications via menus.

kdvi A previewer for the DVI (device independent) files produced by \LaTeX

kfind A utility for searching for lost files

kfloppy A floppy disk formatter

kghostview A previewer for PostScript and PDF files

kmail An application for sending and receiving email

krn The application for reading newsgroups such as `eduni.dcs.cs1`

kvt A terminal window for issuing Linux commands

The `ALT` key can be used in combination with the left mouse button to move windows or with the middle button for bringing to front or sending to back. It can be used with the right mouse button to resize windows.

11 Java

Java programs are compiled and run with a collection of commands called the Java Development Kit (the JDK).

javac *file.java* Compile the *file.java* to produce *file.class*

java *file* Run the class file *file.class*. (Note: do not supply the `.class` extension.)

appletviewer *file.html* View the HTML page in *file.html* which contains an embedded Java applet

12 Compression

The GNU compression utilities reduce the disk space used by a file, allowing the full contents to be recovered later.

gzip *file* Replaces *file* with *file.gz*, a compressed version of the file

gunzip *file.gz* Uncompresses the file to recover the original contents

zgrep *pattern file(s)* Search for a pattern in a list of compressed files

13 Using floppy disks

The MTOOLS collection of utilities provide access to MS-DOS formatted floppy disks from Linux.

mdir *a*: Display the directory listing for the floppy disk in drive *a*:

mcopy *file1 a:file1* Copy *file1* onto floppy disk

mcopy *a:file1 file1* Copy *file1* from floppy disk

14 Command summary

Command	Description	Section
	pipeline	(§7.2.4)
<, >, >>	redirecting I/O	(§7.2.3)
!, !!	recall commands	(§7.3)
%	bring job to foreground	(§7.2.5)
acroread	previewer for PDF files	
alias	abbreviate command	(§7.1)
appletviewer	run Java applet	(§11)
bg	put job in background	(§7.1)
cat	list file	(§4.1.2)
cd	change directory	(§4.2)
chmod	change file permission	(§4.3)
cp	file copy	(§4.1.2)
diff	list file differences	(§8)
du	disk usage	(§4.4)
export	update environment var	(§7.1)
fg	bring job to foreground	(§7.1)
gcc	GNU C compiler	
grep	string searching	(§8)
gunzip	uncompress a file	(§12)
gzip	compress file	(§12)
java	the Java interpreter	(§11)
javac	the Java compiler	(§11)
jobs	list jobs	(§7.2.5)
kdehelp	<i>KDE online help</i>	(§1)
kill, xkill	kill processes	(§6)
logout	finish work session	(§2)
lpq	inspect print queue	(§5)
lpr	print file	(§5)
lprm	remove print job	(§5)
ls	list file names	(§4.1.2)
man, xman	<i>online help</i>	(§1)
mcopy	MSDOS file copying	(§13)
mdir	list MSDOS directory	(§13)
mkdir	make directory	(§4.2)
more	list file pagewise	(§4.1.2)
mv	rename (move) file	(§4.1.2)
ps	list processes	(§6)
quota	disk usage	(§4.4)
rmdir	remove directory	(§4.2)
rm	remove (delete) file	(§4.1.2)
set	update variable	(§7.1)
source	execute shell commands	(§7.1)
w, who	who is logged in	(§6)
yppasswd	change password	(§2)