

UNIX Quick Reference

This card represents a brief summary of some of the more frequently used UNIX commands that all users should be at least somewhat familiar with. Some commands listed have much more functionality than can be included on this card. Please consult their respective manual pages for more information (`man command`).

Filesystem Management

<code>ls</code>	List “normal” files.
<code>ls file</code>	List only <i>file</i> .
<code>ls -a</code>	List all files.
<code>ls -l</code>	Verbosely list “normal” files.
<code>ls -al</code>	Verbosely list all files.
<code>ls -al file</code>	Verbosely list only <i>file</i> .
<code>rm file</code>	Remove <i>file</i> .
<code>rm -r dir</code>	Remove <i>dir</i> and all subdirs.
<code>mv file1 file2</code>	Move/Rename <i>file1</i> to <i>file2</i> .
<code>mv dir1 dir2</code>	Move/Rename <i>dir1</i> to <i>dir2</i> .
<code>cp file1 file2</code>	Copy <i>file1</i> to <i>file2</i> .
<code>cd dir</code>	Change directory to <i>dir</i> .
<code>mkdir dir</code>	Make a directory <i>dir</i> .
<code>rmdir dir</code>	Remove the directory <i>dir</i> .
<code>chown user file</code>	Change owner of <i>file</i> to <i>user</i> .
<code>chgrp group file</code>	Change group of <i>file</i> to <i>group</i> .
<code>chmod nnn file</code>	Change permissions on <i>file</i> to
	755 <code>rwxr-xr-x</code>
	700 <code>rwX-----</code>
	644 <code>rw-r--r--</code>
	600 <code>rw-----</code>
<code>locate file</code>	Find <i>file</i> (fast, not reliable).
<code>find dir -name file</code>	Find <i>file</i> in subdir of <i>dir</i> (very slow, reliable).

Account Management

<code>passwd</code>	Change your password.
<code>chsh</code>	Change your login shell.
<code>chfn</code>	Change your finger info.
<code>who</code>	Show who is logged on.
<code>w</code>	Show who is doing what.
<code>quota -v</code>	Show your disk quota info.
<code>du -k <i>dir</i></code>	Show disk usage of <i>dir</i> .
<code>df -k</code>	Show filesystem usage.

File Manipulation

<code>vi <i>file</i></code>	Edit <i>file</i> with vi editor.
<code>pico <i>file</i></code>	Edit <i>file</i> with pico editor.
<code>wc <i>file</i></code>	Count words/lines in <i>file</i> .
<code>more <i>file</i></code>	Show contents of <i>file</i> .
<code>head -n <i>file</i></code>	Show first <i>n</i> lines of <i>file</i> .
<code>tail -n <i>file</i></code>	Show last <i>n</i> lines of <i>file</i> .
<code>grep <i>pattern file</i></code>	Search <i>file</i> for <i>pattern</i> .
<code>cat <i>file1 file2</i> > <i>file3</i></code>	Append <i>file2</i> to <i>file1</i> and save results as <i>file3</i> .

Process (Job) Mananagement

<code>top</code>	Show the top cpu processes.
<code>ps -elf</code>	Show info about all processes.
<code>kill <i>pid</i></code>	Kill process <i>pid</i> .
<code>kill -9 <i>pid</i></code>	Kill stubborn process <i>pid</i> .
<code>^C</code>	Interrupt a foreground process.
<code>^Z</code>	Suspend a foreground process.
<code>jobs</code>	Show jobs from this session.
<code>bg</code>	Send process to background.
<code>fg</code>	Bring process to foreground.
<code>nice <i>command</i></code>	Run <i>command</i> at lower priority.
<code>renice -n <i>n pid</i></code>	Lower priority of <i>pid</i> by <i>n</i> .
<code>time <i>command</i></code>	Time the execution of <i>command</i> .

Archiving and Compression

<code>tar cf <i>file.tar files</i></code>	Archive <i>files</i> as <i>file.tar</i>
<code>tar xf <i>file.tar</i></code>	Extract archive <i>file.tar</i> .
<code>gzip <i>file</i></code>	Compress <i>file</i> to <i>file.gz</i> .
<code>gzip -d <i>file.gz</i></code>	Uncompresss <i>file.gz</i> .
<code>compress <i>file</i></code>	Compress <i>file</i> to <i>file.Z</i> .
<code>uncompress <i>file.Z</i></code>	Uncompress <i>file.Z</i> .

Printing

<code>lpr file</code>	Print <i>file</i> to default printer.
<code>lpr -Pprinter file</code>	Print <i>file</i> to <i>printer</i> .
<code>lpq</code>	Show default print queue.
<code>lpq -Pprinter</code>	Show <i>printer</i> print queue.
<code>lprm -</code>	Purge default printer.
<code>lprm -Pprinter -</code>	Purge <i>printer</i> .
<code>lprm job</code>	Remove <i>job</i> on default printer.
<code>lprm -Pprinter job</code>	Remove <i>job</i> on <i>printer</i> .

Compilation

(Fortran)	<code>f77 options file1.f ... fileN.f</code>
(C)	<code>cc options file1.c ... fileN.c</code>
(C++)	<code>CC options file1.C ... fileN.C</code>

Some of the more important compiler options are summarised below. Please note that some compilers are slightly different than others, the options included here are common to most compilers.

<code>-o file.x</code>	Create executable called <i>file.x</i> .
<code>-g</code>	Include info for debugging.
<code>-c</code>	Compile, but do not link objects.
<code>-ON</code>	Optimize executable ($1 \leq N \leq 3$).
<code>-lfoo</code>	Include library <i>libfoo.a</i> .

The dbx Debugger

<code>dbx file.x</code>	Start debugger with <i>file.x</i> .
<code>rerun</code>	Rerun from beginning.
<code>stop in sub</code>	Set breakpoint in <i>subroutine</i> .
<code>stop at N</code>	Set breakpoint at line <i>N</i> .
<code>trace var</code>	Set a tracewatch on <i>variable</i> .
<code>status</code>	Show breakpoints and traces.
<code>delete N</code>	Delete <i>N</i> th breakpoint/trace.
<code>delete all</code>	Delete all breakpoints/traces.
<code>cont</code>	Continue running (after a stop).
<code>step</code>	Execute the next line and stop.
<code>next</code>	Same as step , but steps over calls.
<code>list</code>	List 10 lines of source.
<code>list N</code>	List 10 lines from line <i>N</i> .
<code>print var</code>	Print the value of <i>variable</i> .
<code>dump</code>	Print the value of all variables.
<code>quit</code>	Quit the debugger.

Getting Help

`man command` Show the manual for *command*.
`man -k keyword` Search man pages for *keyword*.

I/O Redirection

`< file` Redirect *file* to **STDIN**.
`> file` Redirect **STDOUT** to *file*.
`>! file` Redirect **STDOUT** to *file* (overwrite).
`>> file` Redirect **STDOUT** to *file* (append).
`>& file` Redirect **STDOUT** & **STDERR** to *file*.
`| tee file` Show **STDOUT** and copy to *file*.
`|` Redirect **STDOUT** of one command to **STDIN** of another (ie: `ls | more`).

Environment Variables (csh/tcsh)

`echo $var` Show value of *var*.
`printenv` Show value of all variables.
`setenv var value` Set value of *var* to *val*.
`unsetenv var` Remove the variable *var*.

Aliases (csh/tcsh)

`alias` Show all aliases.
`alias name` Show alias *name*.
`alias name "command"` Create an alias called *name* for *command*.

The vi Editor (Basics)

`<Esc>` Return to command mode.
`:q` Quit.
`:q!` Quit and discard changes.
`:w` Save.
`:w file` Save as *file*.
`:wq` Save and quit.
`:wq!` Save and quit (for readonly files).
`ZZ` Save and quit.
`i` Insert text mode.
`a` Append text mode.
`R` Replace text mode.
`u` Undo last action.
`U` Redo last action.
`.` Repeat last command.

The vi Editor (Editing)

o	Insert new line below cursor.
O	Insert new line above cursor.
r	Replace character.
Nx	Delete N characters.
dNd	Delete N lines.
dw	Delete word (punctuation excluded).
dW	Delete word (punctuation included).
D	Delete here to end of line.
yNy	Yank N lines to buffer.
p	Paste buffer after cursor.
P	Paste buffer before cursor.
J	Join current and next line.

The vi Editor (Movement and Searching)

∅	Go to start of line.
\$	Go to end of line.
G	Go to last line.
Ng	Go to N th line.
^f	Page down.
^b	Page up.
%	Find matching brackets.
/pattern	Search forward for <i>pattern</i> .
?pattern	Search backward for <i>pattern</i> .
n	Go to next <i>pattern</i> .
N	Go back to previous <i>pattern</i> .
:1,\$s/pat1/pat2/g	Replace all <i>pat1</i> with <i>pat2</i> .

The vi Editor (vim Extensions)

^V	Visual block mode.
:center	Center current line.
:left	Left-align current line.
:right	Right-align current line.
:set ruler	Show the cursor position.
:syntax on	Enable syntax highlighting.
:help	Additional help.
:set wrapmargin= n	Wrap line when $< n$ columns remain.

