

Cheat sheet: Introduction to Bash For Linux and Mac OS, Page 1

About Filenames:

Toplevel directory: /
Current directory: .
Parent directory: ..
Home directory: ~

Quoting special characters:

Backslash for a single character:
"file\ with\ spaces"
Single quotes escape everything: 'f*?'

cd change working directory. Without options: go to home directory

cd dir change into **dir**

cd .. change to parent

ls list contents of current directory

ls dir list contents of **dir**

ls -l list in long format

ls -a list all files

ls -R recursively list files in subdirectories

ls -d don't go into subdirectories, just list them

ls -S list by size

ls -t list by modification date

man cmd get help for command **cmd**

touch f if **f** exists: update modification date. Otherwise create a new empty file **f**

cp copies files

cp a b copy file **a** to **b**

cp a b c dir/ copy files **a b c** into **dir/**

cp -R old new recursively copies directory **old** into **new**

cp -i a b ask before overwriting files

mv moves files

mv a b move file **a** to **b**

mv a b c dir/ move files **a b c** into **dir/**

mv -i a b ask before overwriting files

rm removes files

rm a remove file **a**

rm -r dir/ recursively delete directory **dir** and all its contents

rm -i a ask before removing files

mkdir d create directory **d**

rmdir d remove directory **d** (only works on empty directories)

Cheat sheet: Introduction to Bash For Linux and Mac OS, Page 2

cat f	write f to screen
less f	display contents of f , with paging keys: space for next page, b goes up, q for exit, / to search
open f	open file with associated program (Mac OS only) Linux alternative depends on installation
reset	fix terminal when it gets confused by binary data

Wildcards are replaced by bash by matching filenames

*	matches any string *txt matches all .txt files a* matches all files starting with a
?	matches a single character doc_v?.txt matches doc_v1.txt, doc_v2.txt, doc_va.txt etc.
[ac5]	matches one of a, c, or 5
[a-z]	matches a lowercase letter
[a-zA-Z]	matches any letter
[0-9]	matches any digit
[^0-9]	carets inverts meaning: this matches any character that is not a digit

Braces use this to generate strings

c{a,u}t	expanded to "cat cut"
c{1..4}t	range: expanded to "c1t c2t c3t c4t"

Tip: use the "echo" command to try out wildcards/braces.

>, >>	output redirection: send output to a file
ls > f	saves output to file f . If it exists, f will be overwritten
ls >> f	appends output to file f .
<	Input redirection: get input from file
grep x < file	equivalent to "grep x file"
tr a b < old > new	get the input for tr from file old and save output to new . this is necessary because tr does not accept a filename argument.
 	pipe: redirect output from one program to input of another program
ls grep hello	puts output of ls through grep
\$()	put output of command on command line (command substitution)
cat \$(ls -rt tail -n1)	The part in braces outputs the filename of the last modified file. cat will get that filename as its argument
;	put multiple commands on a single line
touch a; ls	first run touch , then ls .

Cheat sheet: Introduction to Bash For Linux and Mac OS, Page 3

Keys

Ctrl-D	Exit	Ctrl-a	Start of line	Ctrl-D	Delete 1 char
Ctrl-L	Clear screen	Ctrl-e	End of line	Ctrl-H	Backspace
Ctrl-C	End process	Ctrl-f	Forward 1 char	Alt-D	Delete a word
Ctrl-Z	Suspend process	Ctrl-b	Back 1 char	Ctrl-W	Delete word backwards
Up or Ctrl-p	History back	Alt-f	Forward 1 word	Ctrl-K	Delete rest of line
Down or Ctrl-n	History fwd	Alt-b	Back q word	Ctrl-U	Delete from start of line
Ctrl-r	Search history				

& start a job in background by ending the line with **&**

fg, bg Put job in foreground/background

When given no arguments, **fg** and **bg** act on current job

fg %2 send job with id 2 to foreground

fg %vi send job with process name "vi" to foreground

kill kill a process

kill %1 kill by job id

kill 12345 kill by process id

kill -KILL 12345

"hard" kill process id (when it doesn't react to normal signals)

xkill kill a process by clicking its windows (X Windows only)

pkill kill process by matching filenames

ps list processes

ps -ef list all processes in long format (Linux)

ps aux list all processes in long format (Mac OS)

top list and manage top processes

sort sort input. With no arguments: sort alphabetically

-n sort numerically

-r reverse sort

-k2 sort by second column

-k2 -t, sort by second column, delimiter is ","

uniq remove repeated lines from input

-c print repeat counts

Cheat sheet: Introduction to Bash For Linux and Mac OS, Page 4

grep	search text
	Note: grep understands <i>regular expressions</i>
grep x f	prints every line of file f containing the string "x"
grep x *	prints every line of every file in the current directory containing the string "x"
-i	case-insensitive search
-c	print number of matching lines
-v	invert meaning of search: will filter out matching lines
-l	print matching line numbers
wc	word count
wc f	prints number of lines, words, and bytes in f
wc -l f	prints number of lines in f
head	prints first 10 lines of input
head -n 5	prints first 5 lines of input
tail	prints last 10 lines of input
tail -n 5	prints last 5 lines of input
tail -f txt	follows file "txt" and updates when lines get appended to it
tr	replace characters in input
	Note: tr does not take a file name argument, so use redirection to give it input
tr o e < f	replace all o characters in f with e
sed	stream editor
	has many features, only most common use shown here:
sed 's/in/out/g'	replaces every occurrence of "in" in input with "out"
find	search for files in a directory
find dir	lists all files in all subdirectories of dir
find . -name '*.txt'	finds all .txt files under the current directory. Note that the wildcard is escaped because find will match filenames itself.
cut	select columns from input
cut -f 2	prints only second column
-d,	sets delimiter to ,
paste	put lines from input files together in a single line
join	like paste, but matching by a key field

Cheat sheet: Introduction to Bash For Linux and Mac OS, Page 4

Configuration files:

.profile for login shells
.bashrc for non-login shells

Put this line in `.profile` to automatically read `.bashrc`:

source ~/.bashrc

Some environment variables

PATH where bash looks for executables

PS1 the prompt

EDITOR your preferred editor

env print currently defined environment variables

Get a variable's value with `$`:

echo \$PATH

Set it without a dollar, make sure you put no whitespace around the =

PATH="\$PATH:~/bin"

alias set aliases

without arguments: list current aliases

alias ls="ls -F" sets "ls -F" as alias for **ls**

\ls use the "original" command (not the alias) by prefixing it with a backslash