

Linux Cheat Sheet for the Bash Shell

Version: 18:23, February 12, 2024

Robert Michael Lewis

Department of Computer Science

College of William & Mary

Moving around the command line

→	Move the cursor forward one character
←	Move the cursor backwards one character
CTRL-f	Move the cursor forward one character
CTRL-b	Move the cursor backwards one character
ESC-f	Move the cursor forward one word
ESC-b	Move the cursor backwards one word
CTRL-a	Go to start of line
CTRL-e	Go to end of line
CTRL-d	Delete the character at the cursor
ESC-d	Delete the text from the cursor to the end of word
CTRL-u	Delete the text from start of line to cursor
CTRL-k	Delete the text from cursor to end of line

Command history

history	Show command history
↑	Scroll backwards through history
↓	Scroll backwards through history

Bash shortcuts

CTRL-c	Interrupt (kill) current command
CTRL-z	Suspend current command
!!	Repeat last command
!-n	Repeat the command that is n commands back
!n	Repeat command n in the history
!abc	Run last command starting with abc
!abc:p	Print last command starting with abc
!\$	Last argument of previous command
!*	All arguments of previous command
^foo^bar^	Run previous command, replacing first instance of foo with bar
!!:gs/foo/bar/	Run previous command, replacing all instances of foo with bar

Who's there?

w	show who is logged in and system load
whoami	display userid

Directory shorthand

~	Your home directory (“tilde”)
~foo	The home directory of user foo (“tilde foo”)
.	The current working directory (“dot”)
..	The parent directory (“dot dot”)

Directory operations

pwd	Print current working directory
cd dir	Change directory to dir
cd ..	Go up a directory level
ls	List files
pushd dir	cd to dir; push dir onto directory stack
popd	Pop directory off stack; cd to new top directory
dirs	Show the directory stack
mkdir dir	Create directory dir
rmdir dir	Remove director dir (must be empty)
rm -rf dir	Remove dir without asking for confirmation

ls options

-a	Show all, including hidden files
-R	Recursively list all subdirectories
-t	Sort by last modified date
-l	Long listing format
-S	Sort by file size
-h	Use familiar units for file sizes
-1	List one file per line
-r	List in reverse alphabetical order
-m	Comma separated output

Getting help	
man cmd	Show manual page for cmd
type cmd	Show command type (e.g., built-in, alias)

File operations	
touch file	Create file or update its last modified time
cat file1 file2	Concatenate files and print to terminal
less file	View file
file file	Get type of file
cp file1 file2	Copy file1 to file2
cp -r dir1 dir2	Recursively copy directory dir1 to dir2
mv file1 file2	Rename (move) file1 to file2
rm file	Delete (remove) file
head file	Show first 10 lines of file
head -n N file	Show first N lines of file
tail file	Show last 10 lines of file
tail -n N file	Show last N lines of file
tail -F file	Print the last lines of file as it changes
sdiff file1 file2	Side-by-side comparison of two files
uniq file	Print unique lines in file
sort file	Sort lines in file alphabetically
sort -n file	Sort lines in file numerically
cut -f n file	Cut column (field) n from file
wc file	word, line, character, and byte count

Searching for files	
find . -name foo	Recursively search the current directory for file foo
find . -name *.py	Recursively search for filenames ending in .py
find . -name foo*	Recursively search for filenames beginning with foo
find . -name *foo*	Recursively search for filenames containing foo

Searching in files	
fgrep "string" foo	Search for lines containing string in file foo
fgrep -i "string" foo	Ignore case while searching
fgrep -v "string" foo	Search for lines NOT containing string in file foo
fgrep "string" -r .	Recursively search files in current directory
fgrep "string" -r dir	Recursively search files in dir subdirectories

I/O redirection in bash (cmd is a command)	
stdin is standard input from terminal	
stdout is standard output from terminal	
stderr is error output to terminal	
cmd < file	Read stdin for cmd from file
cmd1 <(cmd2)	Execute cmd2; use its output as stdin to cmd1
cmd > file	Write stdout of cmd to file
cmd > /dev/null	Discard stdout of cmd (write to bit bucket)
cmd >> file	Append stdout to file
cmd 2> file	Redirect stderr of cmd to file
cmd 1>&2	Redirect stdout to same place as stderr
cmd 2>&1	Redirect stderr to same place as stdout
cmd &> file	Redirect stdout and stderr of cmd to file

Pipes	
cmd1 cmd2	pipe stdout of cmd1 to stdin of cmd2
cmd1 & cmd2	pipe stderr of cmd1 to stdin of cmd2

Bash shell variables	
echo \$VAR	Print shell variable VAR
VAR=value	Set VAR to value
env	Print all shell variables
printenv	Print all shell variables

I/O redirection (cmd is a command)	
cmd < file	Read the input of cmd from file
cmd1 <(cmd2)	Execute cmd2 and use its output as input to cmd1
cmd > file	Write standard output of cmd to file
cmd > /dev/null	Discard standard output of cmd by writing to the bit bucket
cmd >> file	Append standard output to file
cmd 2> file	Redirect error output of cmd to file
cmd 1>&2	Redirect standard output to the same place as standard error
cmd 2>&1	Redirect standard error to the same place as standard output
cmd &> file	Redirect all output of cmd to file

Pipes	
cmd1 cmd2	send standard output of cmd1 to standard input of cmd2
cmd1 & cmd2	send standard error of cmd1 to standard input of cmd2

Shell variables	
echo \$VAR	Print shell variable VAR
VAR=value	Set VAR to value
env	Print all shell variables
printenv	Print all shell variables

Working remotely	
ssh foo@hostname	Secure shell login to remote machine "hostname" as user "foo"
scp foo@hostname:foo/bar.c .	Secure copy of remote file ~foo/foo/bar.c to the local current working directory
scp bar.c foo@hostname:foo	Secure copy of the local file bar.c to remote directory ~foo
scp -r dir1 foo@hostname:	Recursively copy directory dir1 to remote ~doofus