# Shell Cheat Sheet Jacob Koziej (EE '25) CC BY-NC-SA 4.0

#### File Paths

- / root directory
- . current working directory

... — parent directory of the current working directory

/absolute/path — absolute path

relative/path — relative path in the current working directory ./relative/path — relative path in the current working directory

- ../relative/path relative path in the parent directory Tips:
- absolute paths always start with /

command short flag

• you can chain together relative path components with /

**Command Structure** cat

-A -number hello-world.c

> file argument long flag

-afhlNR

you can chain together short flags

--port 31415

some flags take arguments

-p31415

spaces for short flag aruments aren't mandatory

#### if=/dev/zero

sometimes you'll need to use = to specify an argument

i love cooper union ...

most commands can take more than one argument

#### Navigation

pwd — print name of working directory

**1s** — list directory contents

- 1s -al list all directory contents in the long listing format
- tree list contents of directories in a tree-like format

cd — change the working directory to \$HOME

- cd dir/ change the working directory to dir
- cd — change the working directory to the previous directory
- clear clear the terminal screen

## File Manipulation

 $\verb"touch" grass - create or update timestamps of a file named \verb"grass"$ mkdir dir/ — create a directory named dir mkdir -p path/to/dir/ — create all parent directories up to dir

rmdir dir/ — remove an empty directory named dir

rm foo — remove a file named foo

rm -rf dir/ — recursively and forcefully remove dir

cp src dst — copy src to dst

- cp -r src/ dst recursively copy src to dst
- mv src dst move src to dst
- mv foo dir/ move foo into dir
- ln -s src dst create a symbolic link from src to dst
- **readlink dst** resolve where link **dst** points to

unlink dst — remove link dst

cat foo — print the contents of foo to stdout

- head foo print the first 10 lines of foo to stdout
- tail foo print the last 10 lines of foo to stdout
- grep text foo search foo for instances of text

grep -r text dir/ — recursively search files in dir for text

- wc foo get the word count of foo
- wc -l foo get the line count of foo
- sort foo sort the lines of foo

sort -r foo — sort the lines of foo in reverse Tips:

- you **cannot** restore files deleted with **rm**
- both cp and mv will overwrite an existing destination file
- files don't need extension names
- directories don't need a trailing / when used as an argument • everything is *technically* a file in UNIX-like environments

- IO Streams
- cat foo > out redirect stdout of cat to file out

cat foo >> out — append stdout of cat to file out

cat foo 2> /dev/null — discard stderr of cat

grep text foo | sort — pipe stdout of grep to stdin of sort Tips:

- you can chain multiple commands together with multiple pipes
- you can pipe into tee to view the output of a redirection

#### **File Permissions**

-rwxr-xr			rwx	r-x	r			
= / = / = / = /		I						
			421	401	400	< c	lec r	notation
+> c	other (d	b)	111	101	100	< ł	oin r	notation
			$\_/$	\_/	$\backslash_/$	+		
+> g	group (g	g)		I	Ι	r	<->	read
1 1			v	v	v	w V	<->	write
+> u	ıser (ı	1)	7	5	4	x	<->	execute
1		+				-+ -	<->	denied
+> f	ile tvr	be (reg	rular	fil	e. di	recto	prv.	etc)

chmod 644 foo — change foo's permissions to rw-r--r--

chmod +x foo — set foo's executable bit for everyone

chmod g-w foo — deny writes to foo by users in foo's group

- chown jacob foo change foo's owner to jacob
- chgrp wheel foo change foo's group to wheel

chown jacob:wheel foo — set owner to jacob and group to wheel

#### **Environment Variables**

**\$HOME** — user home directory

**\$PATH** — executable path

echo \$F00 — print the value of F00 to stdout

- env view current environment variables
- export FOO=bar set environment variable FOO to bar
- export PATH="\$HOME/bin:\$PATH" add \$HOME/bin to PATH
- unset FOO unset environment variable FOO
- Tips:

Tips:

- Anything inside single quotes is treated as a string literal
- Some environment variables modify program behavior
- A leading \$ expands a shell variable
- Use \${F00} when expanding next to trailing letters

## Shell Expansions

- expands to the value of <code>\$HOME</code>
- !! previous command
- !string last command starting with string
- **\$?** exit code of the last command
- \$(cat foo) treat the output of cat foo as a string
- \* expand to all files in the current working directory
- foo\* expand to all files that start with foo
- \*.c expand to all files that end with .c

## **Reading the Manual**

**External Resources** 

Spot any errors or have a suggestion?

- man man read man's manual
- man cat read cat's manual
- man stdint.h read stdint.h's manual

When in doubt, read the man page

explainshell.com — explain shell commands

devhints.io/bash — Bash scripting cheatsheet

- man printf read the shell's printf manual
- man 3 printf read libc's printf() manual

man 2 write — read the system's write() manual

• If there's no man page, check a command's -h/--help flag

• Adding a number to man specifies a different section

missing.csail.mit.edu — MIT's Missing Semester of CS

github.com/dylanaraps/pure-sh-bible — pure sh bible

Shoot me an email at <jacobkoziej@gmail.com> or <jacob.koziej@cooper.edu> v0.1.2 — 2022-08-07