Linux Command Line Cheat Sheet

CSCE 313 – Introduction to Computer Systems · Prep for Lab 1 (SEED Labs / Ubuntu Server VM)

1. Navigating the Filesystem

`pwd`	Print current directory (where am I?)
`ls`	List files in current directory
`ls -la`	List all (incl. hidden), long format
`ls -lh`	Long format, human-readable sizes
`cd dir`	Move into dir
`cd ..`	Move up one directory
`cd ~`	Go to home directory
`cd -`	Go to previous directory
`find . -name "*.c"`	Find files by name pattern

2. Files & Directories

`mkdir name`	Create a directory
`mkdir -p a/b/c`	Create nested dirs at once
`touch file`	Create empty file / update timestamp
`cp a b`	Copy file a to b
`cp -r dir1 dir2`	Copy directory recursively
`mv old new`	Move or rename
`rm file`	Delete a file
`rm -r dir`	Delete a directory recursively

rm has no recycle bin – deleted is gone. Double-check paths before rm -rf.

3. Viewing & Reading Files

`cat file`	Print entire file to screen
`less file`	Page through file (q: quit, /: search)
`head -n 20 file`	Show first 20 lines
`tail -n 20 file`	Show last 20 lines
`tail -f file`	Follow file live (e.g. a log)
`wc -l file`	Count lines in a file

4. Searching & Text Processing

`grep "pat" file`	Search for text matching pattern
`grep -rn "pat" dir`	Recursive search, show line numbers
`grep -i "pat" file`	Case-insensitive search
`diff f1 f2`	Show differences between two files
`sort file`	Sort lines alphabetically
`man command`	Open the manual page for a command

5. Permissions & Ownership

`ls -l`	Shows `-rwxr-xr-x owner group`
`chmod +x script.sh`	Make a file executable
`chmod 755 file`	Set perms numerically (rwx r-x r-x)
`chmod u+w file`	Add write perm for owner (symbolic)
`chown user:group file`	Change file owner/group

r=4, w=2, x=1 per group (owner/group/other) – add them up, e.g. 755 = rwx r-x r-x.

6. Processes & System Monitoring

`ps aux`	List all running processes
`top`	Live process/CPU monitor (q: quit)
`htop`	Nicer interactive process monitor
`kill PID`	Ask a process to terminate
`kill -9 PID`	Force-kill a process
`df -h`	Disk space per filesystem
`du -sh dir`	Total size of a directory
`free -h`	Memory (RAM) usage

7. Compiling C Code

`gcc -o prog file.c`	Compile file.c into executable prog
`gcc -Wall -g -o prog file.c`	...with warnings + debug symbols
`./prog arg1 arg2`	Run the compiled binary
`gcc -c file.c`	Compile to object file (file.o) only
`gcc a.o b.o -o prog`	Link object files into an executable
`make`	Build using the project's Makefile
`make clean`	Remove build artifacts (per Makefile)

-g embeds debug symbols – required for GDB to show source lines and variable names.

8. Archives & File Transfer

`tar -czvf out.tar.gz dir/`	Create a gzip-compressed archive
`tar -xzvf out.tar.gz`	Extract a .tar.gz archive
`unzip file.zip`	Extract a .zip archive
`scp file user@host:path`	Copy a local file to a remote host
`scp -r dir user@host:path`	Copy a directory to a remote host
`scp user@host:path .`	Copy a remote file to local machine

9. Git Basics

`git clone url`	Clone a repository
`git status`	Show changed / staged files
`git add file`	Stage a file for commit
`git add .`	Stage all changes in this directory
`git commit -m "msg"`	Commit staged changes
`git push`	Upload commits to the remote
`git pull`	Download and merge remote changes
`git log -oneline`	Compact commit history

10. SSH & Remote Access

`ssh user@host`	Connect to a remote machine
`ssh -p 2222 user@host`	Connect using a custom port
`ssh-keygen -t ed25519`	Generate a new SSH key pair
`ssh-copy-id user@host`	Install your public key on a host
`exit`	Close the SSH session

VS Code Remote-SSH uses the same key/config as your terminal ssh – test with ssh first.

11. GDB Quick Reference (Lab 1)

`gdb ./prog`	Start GDB on a compiled binary
`run arg1 arg2`	Start the program (with args) under GDB
`break funcname`	Set breakpoint at a function
`break file.c:42`	Set breakpoint at a line number
`next / n`	Step over (skip into function calls)
`step / s`	Step into a function call
`continue / c`	Resume until the next breakpoint
`print var / p var`	Print the current value of a variable
`backtrace / bt`	Show the current call stack
`watch var`	Break whenever var's value changes
`list`	Show source code around current line
`quit / q`	Exit GDB

Terminal Survival Tips

`Tab`	Auto-complete file/dir names and commands
`↑ / ↓`	Cycle through command history
`Ctrl+C`	Interrupt / kill the running command
`Ctrl+D`	Send EOF – exits shell or a program's input
`Ctrl+R`	Search command history
`clear`	Clear the terminal screen
`command -help`	Quick usage summary for a command

Relative paths (./file, ../dir) are read from your current directory – run pwd if unsure.