

Reading 00

★ The Linux Command Line (book) ★

★ Chapter 1: What is the shell?

Shell: a program that takes commands from the keyboard and gives them to the operating system to perform

• On most Linux systems a program called bash acts as the shell program.

↳ other shell programs include ksh, tcsh, zsh.

terminal: a program that opens a window and lets you interact with the shell.

★ Chapter 2: Navigation

root directory: the 1st directory in the file system.

Commands

"pwd" → print working directory

"cd .." → go to parent directory

"cd" → will go to home directory

"cd -" → changes the working directory to the previous one

• File names that start with a period character are hidden.

↳ "ls" will NOT list these files

↳ "ls -a" will list these files

★ Chapter 3: Looking Around

- "ls" → list files and directories
- "less" → view text files
- "file" → classify a file's contents
- "ls -l" → long list files and directories
- "ls -la" → long list ALL files (even hidden ones)

- Most commands operate like this:

Command - options arguments

- Displaying a text file:

less text-file

Controlling less

Command	Action
Page Up or b	Scroll back one page
Page Down or space	Scroll forward one page
G	Go to the end of the text file
G	Go to the beginning of the text file
/characters	Search forward in the text file for an occurrence of the specified characters
n	Repeat previous search
h	Display a complete list less commands and options
q	Quit

- "file" will examine the file and tell us what kind of file it is.

file name_of_file

★ Chapter 4: A guided tour

- Symbolic links are a special type of file that points to another file.
 - ↳ With symbolic links it is possible for a single file to have multiple names.
 - ↳ To create symbolic links, we use the "ln" command.

★ Chapter 5: Manipulating Files

Wildcard	Meaning
*	Matches any characters
?	Matches any single character
[characters]	Matches any character that is a member of the set of characters
[!characters]	Matches any character that is NOT a member of the set of characters.

- Copying multiple files into a directory:

cp file1 file2 ... directory

- Overwriting contents of file2 with contents of file1

cp file1 file2

Command: mv file1 file2

- If file2 does NOT exist, then file1 is renamed file2.
- If file2 DOES exist, its contents are silently replaced with the contents of file1.

Command: mv dir1 dir2

- If dir2 does NOT exist, then dir1 is renamed dir2
- If dir2 DOES exist, then directory dir1 is moved within directory dir2.

Command: rm file ...

- removes all files specified

Command: rm -r directory ...

- removes all specified directories

★ Chapter 6: Working with Commands

- "type" → Display info about command type
- "which" → Locate a command
- "help" → Display reference page for shell builtin
- "man" → Display an on-line command reference

- Commands can be one of 4 different kinds:
 1. An executable program
 2. A command built into the shell itself
 3. A shell function
 4. An alias

- "which" is used to determine the exact location of a given executable
 - ↳ only works for executable programs.

- "help -m cd" → gives docs on cd

↳ in docs, square b ts indicate optional items, and a pipe indicates mutually exclusive items.

- "mkdir --help" → displays a description of the command's supported syntax and options.

- "man -ls" → displays manual for ls, you can use this for reference.

★ A beginners guide to Git version control ★

- Files that have been staged are ready to be committed to the repository you are working on.

- "git init" → initializes git repo

- "git status" → shows status of repo

- "git add file.txt" → adds file.txt to staging area

- "git rm --cached file.txt" → maintains file in working directory while removing it from the Git index.

- "git add ." → adds all files to staging area

- Git uses commits to make changes to files and directories permanent.

↳ Every commit represents a new version of our repository.

- "git commit -m "my first git commit" " → committing everything in staging area

- "git log" → shows history of changes

- "git push" → uploads content from local repo to remote repo

- or "git push origin branchname"

★ Git Handbook ★

- A repo encompasses the entire collection of files and folders associated with a project, along with each file's revision history.
- "git clone" → creates a local copy of a project that already exists remotely.
- "git branch" → shows the branches being worked on locally.
- "git merge" → merges lines of development together
- "git pull" → updates the local line of development with updates from its remote counterpart.
- "git push" → updates the remote repository with any commits made locally to a branch.
- "git branch my-branch" → creates a new branch
- "git checkout my-branch" → switches to that branch

• Models for collaborative development :

1. Shared repo

- individuals and teams are explicitly designated as contributors with read, write, or administrator access

2. Fork and Pull

- for open-source projects
- A fork is a copy of a project under a developer's personal account.
- Work completed in forks is either kept separate, or is surfaced back to the original project via pull request.