Linux Command Line Course Notes

Introduction

Linux is a powerful, open-source operating system widely used in servers, development, and DevOps. The command line (CLI) is essential for interacting with Linux efficiently.

Benefits of Using the Command Line

- Greater control than GUI
- Automation with scripts
- Consistent across distributions

---

1. Setting Up Your Environment

Creating a Linux Virtual Machine

- Use **VirtualBox**, **VMware**, or **WSL (Windows Subsystem for Linux)**
- Install a distribution (e.g., **Ubuntu**, **CentOS**)
- Allocate resources (RAM, CPU, Disk)
- Use snapshot feature for safe experimentation

Alternatives to a VM

- **MacOS/Linux host OS:** Direct terminal use
- **Windows:** Git Bash, WSL
- **Online terminals:** Katacoda, Replit

Using GitHub Codespaces

- Provides a **cloud-based Linux environment**
- No local setup required
- Useful for quick command trials and saving work

---

2. Command-Line Basics

What is the Command Line

- A **text-based interface** to interact with the OS
- Common shells: **bash**, **zsh**, **fish**
- Command prompt format:

- `username@hostname:~$`

Command Structure

- Syntax:
- `command [options] [arguments]`
- Example:
- `ls -l /home`

Running Commands

- Execute line by line
- Press `Enter` to run
- Use `Ctrl+C` to stop a running process

Getting Help

- `man <command>` — view manual page for a command
- `<command> --help` — quick help and usage options
- `info <command>` — detailed documentation (often more verbose than `man`)
- `apropos <keyword>` — search man pages for commands related to a keyword
- `help <builtin>` — help for shell built-in commands
- `whatis <command>` — brief one-line description of a command
- `which <command>` — show the full path of the command binary
- `type <command>` — tell whether a command is a binary, alias, or shell built-in
- `whereis <command>` — show binary, source, and man page locations
- `man -k <keyword>` — search for commands by keyword (same as `apropos`)
- `man -f <command>` — show a one-line description of a command (same as `whatis`)
- `tldr <command>` — simplified examples of how to use a command (if installed)

Useful Keyboard Shortcuts

- `Ctrl+C` — cancel/terminate the current process
- `Ctrl+Z` — suspend/stop the current process (resume with `fg`/`bg`)
- `Ctrl+D` — logout / end-of-file (EOF) / exit shell
- `Ctrl+L` — clear the terminal (same as `clear`)
- `Ctrl+A` — move cursor to the beginning of the line
- `Ctrl+E` — move cursor to the end of the line
- `Ctrl+U` — cut text from cursor to beginning of line
- `Ctrl+K` — cut text from cursor to end of line
- `Ctrl+Y` — paste (yank) the last cut text

- `Ctrl+W` — cut the word before the cursor
- `Alt+D` — cut the word after the cursor
- `Alt+Backspace` — delete the word before the cursor
- `Ctrl+T` — swap the last two characters typed (transpose)
- `Alt+T` — transpose (swap) the last two words typed
- `Alt+U` — uppercase the word after cursor
- `Alt+L` — lowercase the word after cursor
- `Alt+C` — capitalize the word after cursor
- `Ctrl+R` — reverse search through command history
- `Ctrl+S` — forward search through command history (sometimes disabled by default)
- `↑/↓` — cycle through command history
- `!!` — repeat the last command
- `!<number>` — run a specific command from history by its number
- `Tab` — autocomplete file/command name
- `Tab Tab` — list all possible completions
- `Ctrl+_` — undo last editing command in terminal line

---

3. Files, Directories, and Permissions

Linux File System Structure

- **Root directory** `/` is the top
- Standard folders:
  - `/home` — user files
  - `/etc` — configuration
  - `/bin` — essential commands
  - `/var` — logs, variable data
  - `/tmp` — temporary files
  - `/usr` — user applications
  - `/opt` — optional software packages
  - `/dev` — device files
  - `/proc` — system and process info

File Paths

- **Absolute path:** starts from `/` (e.g., `/home/user/docs`)
- **Relative path:** relative to current dir (e.g., `./docs`)

Navigation Commands

- `pwd` — print working directory
- `cd <dir>` — change directory
- `cd ~` — go to home directory

- `cd -` — go to previous directory
- `ls` — list files
- `tree` — display directory structure (if installed)

Listing Files

- `ls -l` — long format
- `ls -a` — includes hidden files
- `ls -lh` — human-readable sizes
- `ls -lt` — sort by modification time
- `ls -R` — recursive listing

Directory Management

- `mkdir newdir` — create directory
- `mkdir -p dir/subdir` — create nested directories
- `rmdir emptydir` — remove empty directory
- `rm -r dir` — remove directory recursively

File Operations

- `touch file` — create empty file / update timestamp
- `cp file1 file2` — copy
- `cp -r dir1 dir2` — copy directory
- `mv file1 file2` — move/rename
- `rm file` — remove file
- `rm -i file` — remove with confirmation
- `rm -f file` — force remove

Finding Files

- `find /path -name "filename"` — search by name
- `find . -type f -name "*.txt"` — search for files by extension
- `find . -size +10M` — files larger than 10MB
- `find . -mtime -1` — files modified in last 1 day
- `locate filename` — search using database (updated with `updatedb`)

User Roles and `sudo`

- **root (superuser)** and regular users
- `sudo` — run command as root
- `sudo apt update`
- `su -` — switch to root user
- `whoami` — show current user

File Permissions

- **Types:** read (r), write (w), execute (x)
- `ls -l` shows permissions: `-rw-r--r--`
  - First char = type (`-` file, `d` dir, `l` symlink)
  - Next = owner/group/others permissions

Modifying Permissions and Ownership

- `chmod 755 file` — set permissions
- `chmod u+x file` — add execute for user
- `chown user file` — change owner
- `chown user:group file` — change owner and group

Links

- Hard link:
- `ln file linkname`
- Symbolic (soft) link:
- `ln -s file symlink`

---

4. Common Command-Line Tasks and Tools

Unix Philosophy

- Small programs that do one thing well
- Combine with pipes for powerful workflows

Pipes

- `|` passes output of one command to another
- `ls -l | grep ".txt"`

Viewing Text Files

- `cat file` — view whole file
- `tac file` — view file in reverse
- `head -n 10 file` — first 10 lines
- `tail -n 10 file` — last 10 lines
- `tail -f logfile` — live view of log file
- `less file` — scrollable view

Searching Text

- `grep "pattern" file`
- `grep -i "pattern" file` — ignore case
- `grep -r "pattern" dir` — recursive search
- `grep -n "pattern" file` — show line numbers
- `egrep "regex" file` — extended regex

Text Manipulation

- `awk '{print $1}' file` — print first column
- `awk -F, '{print $2}' file.csv` — use custom delimiter
- `sed 's/old/new/g' file` — replace text globally
- `sed -n '5,10p' file` — print lines 5–10
- `sort file` — sort lines alphabetically
- `sort -n file` — numeric sort
- `uniq file` — remove duplicates
- `wc -l file` — count lines

Editing Text

- **Vim:**
- `vim file`
  - Insert: `i`
  - Save & quit: `:wq`
  - Quit without saving: `:q!`
- **Nano:**
- `nano file`
  - Save: `Ctrl+O`
  - Exit: `Ctrl+X`

Archives

- Create tar:
- `tar -cvf archive.tar files/`
- Extract tar:
- `tar -xvf archive.tar`
- Zip:
- `zip file.zip file1 file2`
- Unzip:
- `unzip file.zip`

- Gzip:
-
  ```
  gzip file
  gunzip file.gz
  ```

Output Redirection

- Overwrite:
- `command > file`
- Append:
- `command >> file`
- Input from file:
- `command < file`
- Redirect errors:
- `command 2> error.log`
- Redirect all output:
- `command &> file`

Environment Variables and PATH

- View all:
- `printenv`
- Echo variable:
- `echo $PATH`
- Set variable:
- `export VAR=value`
- Add path:
- `export PATH=$PATH:/new/dir`

---

5. Advanced Topics (A Peek)

System Information

- Distribution:
- `cat /etc/os-release`
- Kernel:

- `uname -a`
- Uptime:
- `uptime`
- Logged-in users:
- `who`
  `w`

Hardware and Disk Info

- CPU:
- `lscpu`
- Block devices:
- `lsblk`
- Disk space:
- `df -h`
- Memory:
- `free -h`
- PCI devices:
- `lspci`
- USB devices:
- `lsusb`

Package Management

Debian/Ubuntu:

```
sudo apt update
sudo apt install <package>
sudo apt remove <package>
dpkg -l        # list installed
```

RedHat/CentOS:

```
sudo yum install <package>
sudo yum remove <package>
rpm -qa        # list installed
```

Process Management

- `ps aux` — show all processes
- `top` — interactive process monitor
- `htop` — improved process monitor (if installed)
- `kill <PID>` — terminate process
- `kill -9 <PID>` — force kill
- `jobs` — list background jobs
- `fg %1` — bring job to foreground
- `bg %1` — continue job in background

Networking

- `ping host` — test connectivity
- `curl url` — fetch content
- `wget url` — download file
- `ifconfig` or `ip a` — show network interfaces
- `netstat -tulnp` — show listening ports
- `ss -tulnp` — modern replacement for netstat

User Management

- `adduser user` — add new user
- `passwd user` — change password
- `deluser user` — delete user
- `id user` — show UID/GID
- `groups user` — show group membership
- `usermod -aG group user` — add user to group

System Monitoring

- `dmesg | less` — kernel logs
- `journalctl -xe` — systemd logs
- `uptime` — system load
- `vmstat` — performance stats
- `iostat` — disk I/O stats