

---

## Topic 1: The Linux Community and a Career in Open Source

---

### 1.1 Linux Evolution and Popular Operating Systems

- Linux Distributions:
  - Debian: A stable and versatile Linux distribution that forms the base for other distributions like Ubuntu.
  - Ubuntu (LTS): A popular, user-friendly Linux distro derived from Debian, focused on long-term support releases.
  - CentOS: Community version of Red Hat Enterprise Linux, known for stability.
  - Red Hat: A commercial distribution widely used in enterprise environments.
  - openSUSE and SUSE: Open-source and enterprise-focused distributions, respectively.
  - Linux Mint: Designed for ease of use and **derived from Ubuntu/Debian**.
  - Scientific Linux: Tailored for scientific and academic use.
  - OpenSUSE
- Embedded Systems:
  - Raspberry Pi: A small, affordable computing platform for learning and experimentation.
  - Raspbian (now Raspberry Pi OS): A Debian-based OS optimized for Raspberry Pi hardware.
  - Android: A Linux-based operating system for mobile and embedded devices.
- Linux in the Cloud:
  - Dominates cloud computing due to scalability, flexibility, and open-source nature.
  - Popular in major cloud platforms like AWS, Azure, and Google Cloud.

---

### 1.2 Major Open Source Applications

- Desktop Applications:
  - LibreOffice: Office suite for word processing, spreadsheets, and presentations.
  - Thunderbird:
    - Email client with calendar and task integration.
    - Access and manage emails here
    - Replacement for Microsoft Outlook
  - Firefox: A widely-used open-source web browser.
    - Can be opened using the command: `firefox`
  - GIMP: Advanced image editing software, an open-source alternative to Photoshop.
  - VLC: Media Player for video and music files
  - Impress: Replacement for Microsoft Powerpoint
  - PostFix: Mail transfer service

- Inkscape: A vector graphics editor that can handle graphical line drawings.
  - ownCloud: self hosting file service
  - Server Applications:
    - Apache HTTPD and NGINX: Leading web servers used to host websites.
      - Listens for network requests and responds to them
    - MariaDB and MySQL: Popular database management systems for web applications.
    - Samba: Used for file sharing between Linux and Windows systems.
  - Development Languages:
    - Popular programming languages supported on Linux: C, Java, Python, PHP, Perl, and shell scripting.
  - Package Management Tools:
    - dpkg and apt-get: For managing Debian-based distributions like Ubuntu.
    - rpm and yum: For managing Red Hat-based distributions like CentOS.
- 

### 1.3 Open Source Software and Licensing

- Open Source Philosophy:
  - Encourages collaboration, transparency, and community-driven development.
- Open Source Licensing:
  - Copyleft: Requires derivative works to be open-sourced (e.g., GPL).
    - General Public License (GPL) guarantees users the freedom to run, study, share, and modify the software
  - Permissive: Allows proprietary code use (e.g., BSD, MIT licenses).
  - Free Software License: Gives conditions for modifying and distributing the licensed software
- Organizations:
  - Free Software Foundation (FSF): Advocates for free software.
  - Open Source Initiative (OSI): Promotes and protects open-source software.
- Value Proposition
  - Security
  - Affordability
  - Transparency
  - Interoperability
  - Scalability
  - Localization
- Key Terms:
  - FOSS (Free and Open Source Software): Software that is free to use, modify, and distribute.
  - FLOSS (Free/Libre Open Source Software): Highlights both "free as in freedom" and "free as in no cost."

---

## 1.4 ICT Skills and Working in Linux

- Desktop Skills:
  - Using browsers, managing privacy settings, and saving web content.
    - Private Browsing - History is not stored locally
  - Creating documents and presentations with open-source tools like LibreOffice.
- Command Line Skills:
  - Navigating and executing commands in the terminal.
  - Understanding password and privacy tools.
- Industry Uses of Linux:
  - Widely used in cloud computing for scalability and cost-effectiveness.
  - Supports virtualization technologies like Docker and Kubernetes.
- Terms
  - tty
    - The UNix name for a physical or virtual terminal connection
- Desktop Environments
  - a graphical user interface (GUI) that provides a visual way for users to interact with the Linux operating system
  - Popular Ones Include:
    - Cinnamon, Gnome, KDE, XFCE, Unity, and Mate
- Virtual Appliance
  - A pre-integrated, self contained system that is made by combining a software application with just enough OS for it to run optimally on industry standard hardware or a virtual machine.

---

## Topic 2: Finding Your Way on a Linux System

---

### 2.1 Command Line Basics

- Basic Shell:
  - The bash shell is the most common command-line interface for Linux systems.
- Command Line Syntax:
  - Commands follow a format of <command> [options] [arguments].
- Variables
  - Follow the format of:
    - variable\_name = \$(command)
      - Or
    - variable\_name = 'command'
- Environment Variables (MUST BE IN CAPS):
  - PATH: Specifies directories where the system looks for executables.

- Allows for commands to be executed without knowing where they are located
  - export: Adds or modifies environment variables.
- Quoting:
  - Use " " (double quotes) to preserve spaces but allow variable expansion.
  - Use ' ' (single quotes) to preserve literal text.
  - Use \ (backslash) to escape special characters.
- Grep
  - Character Classes with Square Brackets []
    - Match specific characters:
      - Example: [aeiou] matches any vowel.
    - Character ranges:
      - [a-z]: Matches any lowercase letter.
      - [A-Z]: Matches any uppercase letter.
      - [0-9]: Matches any digit.
      - [A-Za-z0-9]: Matches alphanumeric characters.
    - Negation:
      - [^xyz]: Matches any character except x, y, or z.
  - Regular Expressions
    - Basic Regular Expressions (BRE)
      - Anchors:
        - ^: Matches the start of a line.
          - Example: grep '^Hello' filename
        - \$: Matches the end of a line.
          - Example: grep 'world\$' filename
      - Dot .:
        - Matches any single character.
          - Example: grep 'h.t' filename matches hat, hot, etc.
      - Asterisk \*:
        - Matches zero or more occurrences of the preceding character.
          - Example: grep 'lo\*ng' filename matches lng, long, loooong.
      - Escape Sequences \:
        - Used to interpret special characters literally.
          - Example: grep '\\$50' filename matches \$50.
    - Extended Regular Expressions (ERE)
      - Enable with grep -E or egrep.
      - Or |:
        - Matches one pattern or another.

- Example: `grep -E 'cat|dog' filename`
- Parentheses ():
  - Group patterns.
    - Example: `grep -E '(cat|dog)s?'` matches cat, cats, dog, or dogs.
- Plus +:
  - Matches one or more occurrences of the preceding character.
    - Example: `grep -E 'go+gle' filename` matches gogle, google, etc.
- Question Mark ?:
  - Matches zero or one occurrence of the preceding character
    - Example: `grep -E 'colou?r' filename`
    - matches color and colour.
- Fixed-String Search
  - Use `grep -F` or `fgrep` to search for patterns as is (no regex interpretation)
    - Example: `grep -F '[abc]' filename` matches the literal [abc]
- Case Sensitivity
  - Ignore case:
    - Use `grep -i`
      - Example: `grep -i 'pattern' filename`
- Whole Word Search
  - Use `grep -w` to match only whole words
    - Example: `grep -w 'word' filename`
- Line Numbers and Context
  - Show line numbers:
    - Use `grep -n`.
      - Example: `grep -n 'pattern' filename`.
  - Show surrounding lines (context):
    - `-A N`: Show N lines after the match.
    - `-B N`: Show N lines before the match.
    - `-C N`: Show N lines before and after the match.
      - Example: `grep -C 2 'pattern' filename`.
- Inverted Search
  - Use `grep -v` to exclude lines matching the pattern.
    - Example: `grep -v 'error' filename`.
- Count Matches
  - Use `grep -c` to count lines that match the pattern.
    - Example: `grep -c 'pattern' filename`.

- Match Only Entire Lines
  - Use `grep -x` to match lines where the entire line matches the pattern.
    - Example: `grep -x 'complete line' filename.`
- File Names
  - Search recursively:
    - Use `grep -r` to search in files within a directory recursively.
      - Example: `grep -r 'pattern' /path/to/directory.`
  - Show only file names:
    - Use `grep -l`.
      - Example: `grep -l 'pattern' filename.`
- Binary Files
  - Search binary files:
    - Use `grep --binary-files=text`.
      - Example: `grep --binary-files=text 'pattern' binaryfile.`
  - Suppress binary output:
    - Use `grep -I`.
      - Example: `grep -I 'pattern' filename.`
- Additional Options
  - Word Boundaries:
    - Use `\<` and `\>` to match the beginning and end of a word.
      - Example: `grep '\<word\>' filename.`
  - Quiet Mode:
  - Use `grep -q` to suppress output and return exit status only.
  - `-c` to clear a users history

---

## 2.2 Using the Command Line to Get Help

- Help Systems:
  - `man`: Manual pages for detailed documentation of commands.
    - Have a Name, Config, and Return value section
    - `man` Options
      - `-k <keyword>`
        - search all man pages for keyword
  - `info`: Provides more detailed and structured documentation.
  - `locate`: Quickly finds files by name on the system.
  - `which`: searches for executable programs or scripts located in the `PATH` variable
  - `/usr/share/doc/`: Directory containing system documentation.
  - `whereis`: lets users locate the executable, source, and man page files for any command

- apropos: searches man page names and description for a user supplied keyword
- 

## 2.3 Using Directories and Listing Files

- Files and Directories:
    - Files and directories are case-sensitive in Linux.
    - Hidden files: Start with a dot (.), viewable with ls -a.
      - Hidden files are created by adding a "." before the filename
        - Example
          - example.txt -> .example.txt
  - Home Directories:
    - Each user has a home directory (e.g., /home/username).
    - Shortcut: ~ represents the home directory.
  - Navigation:
    - cd: Changes the current directory.
    - . and ..: Represent the current and parent directories, respectively.
  - Listing Files:
    - ls: Lists files and directories.
      - Common options: -l for long format, -a for hidden files, -R for recursive listings.
- 

## 2.4 Creating, Moving, and Deleting Files

- File Operations:
    - mv: Moves or \* **renames files** \*
    - cp: Copies files.
    - rm: Deletes files.
    - touch: Creates an empty file.
  - Directory Operations:
    - mkdir: Creates directories.
    - rmdir: Removes empty directories.
  - Key Features:
    - File and directory names are case-sensitive.
    - Use globbing (e.g., \*, ?) for pattern matching.
      - ? is used to represent a single character before or after its appearance
      - \* is used to represent any amount of character before or after its appearance
- 

## Topic 3: The Power of the Command Line

---

### 3.1 Archiving Files on the Command Line

- Files and Directories:

- Archiving allows grouping multiple files into one for easy storage or transfer.
  - Common Tools:
    - tar: Creates archive files.
      - Common options:
        - -c: Create an archive.
        - -x: Extract an archive.
        - -v: Verbose output (shows progress).
        - -f: Specify the file name for the archive.
      - Example: tar -cvf archive.tar file1 file2.
    - Compression Tools:
      - gzip, bzip2, xz: Compress archives.
        - Gzip
        - Bzip2 - Used to compress a single file using the Burrows Wheeler algorithm
      - zip/unzip: Compress and extract files in .zip format.
- 

### 3.2 Searching and Extracting Data from Files

- Search Tools:
  - grep: Searches for patterns in files.
    - Example: grep "pattern" file.txt.
  - cat: allows the creation of single or multiple files, view file contents, concatenate files, and redirect output in the terminal to a file.
  - less, head, tail: Display file contents with control over what to view.
    - Less
    - Head
    - Tail
      - -c to print the last X bytes of a file to the screen
- Sorting and Extracting:
  - sort: Sorts file content line by line.
  - cut: Extracts specified columns or fields from a file.
  - wc: Counts words, lines, and characters in a file.
- Regular Expressions (Regex):
  - Used for pattern matching.
    - Examples:
      - .: Matches any character.
      - [ ]: Matches a set of characters.
      - \*: Matches zero or more repetitions.
      - ?: Matches zero or one repetition.
- Redirection using >
  - Happens first so:



- `sort data.txt > data.txt`
    - Will not return anything because the file text is transferred to the new file already
- 

### 3.3 Turning Commands into a Script

- Shell Scripting Basics:
    - A script automates repetitive command-line tasks.
    - Start scripts with a shebang (`#!/`) to specify the interpreter (e.g., `#!/bin/bash`).
  - Key Elements:
    - Variables: Store values for reuse.
      - Example: `name="Linux"`.
    - Arguments: Passed to scripts using `$1`, `$2`, etc.
      - `$1` would be the first command line argument
      - `$2` would be the second command line argument
    - Loops: Automate repetitive actions.
    - STD
      - STDIN is represented by the file descriptor number 0
      - STDOUT is represented by the file descriptor number 1
      - STDERR is represented by the file descriptor number 2
    - `\d`
      - Option to add the date to the prompt in the shell
    - Discarding errors from command output
      - `command >file 2>/dev/null`
        - 2 represents the error descriptor, where errors are written to
        - `>` redirects it to the right, which is `/dev/null`, the standard Linux device where you send output that you want to be ignored
      - `2>&1` redirects the standard output and standard error to the same file
  - Text Editors:
    - `vi` and `nano`: Common editors for writing scripts.
  - Exit Status:
    - Scripts return an exit code (0 for success, non-zero for errors).
- 

## Topic 4: The Linux Operating System

---

### 4.1 Choosing an Operating System

- Operating Systems:
  - Windows: Proprietary, GUI-focused, limited CLI usage.
  - macOS: Unix-based, GUI-focused, designed for Apple hardware.
  - Linux: Open-source, CLI and GUI support, versatile and customizable.
- Distribution Life Cycle:

- Includes beta versions, stable releases, and end-of-life (EOL) cycles.
  - GUI vs CLI:
    - GUI: User-friendly interface with graphical elements.
    - CLI: Efficient for advanced users, automation, and scripting.
    -
  - Relevant Files, Terms, Utilities:
    - GUI vs CLI, lifecycle management.
- 

## 4.2 Understanding Computer Hardware

- Key Components:
    - Motherboard: Connects all hardware components.
    - Processor (CPU): Executes instructions and processes data.
    - Memory (RAM): Stores data temporarily for fast access.
    - Storage:
      - HDD: Large-capacity magnetic storage.
      - SSD: Faster, solid-state storage.
      - Partitions: Logical divisions of storage devices (/dev/sd\*).
        - Partitions on these disks can range from 1 (for the first partition) to 16 (for the sixteenth partition). If you have a few ide/pata devices installed in a system, they will be hda (the first device), hdb (the second device), hdc (the third device), and so on. Each partition on that device will append a number to it, so the second partition on the first device would be /dev/hda2.
        - Modern software uses sda, sdb since HDDs are outdated
      - Optical Drives
        - CD - 700MB
        - DVD - 4.7GB
  - Peripherals:
    - Input/output devices like keyboards, mice, and monitors.
  - Drivers:
    - Software that allows the OS to interact with hardware.
  - Architecture
    - Know what version of Linux to be installed with hardware
      - 32 Bit = x86
        - uses 4GB RAM max
      - 64 Bit = x64
- 

## 4.3 Where Data is Stored

- Storage Locations:
  - Programs and Configuration:

- /etc/
    - Stores config files in a Linux system
  - /usr/bin/
- Processes:
  - /proc/
    - The /proc tree originated in System V Unix, where it only gave information about each running process, using a /proc/\$PID/stuff scheme. Linux greatly extended that, adding all sorts of information about the running kernel's status. The intended purpose for the /proc virtual filesystem is to provide an interface into the kernel and its processes.
  - /sys/
    - Interacts with the kernel and system as a whole
- Logs:
  - /var/log/, accessed with tools like syslog or dmesg.
    - syslog: system logs
    - dmesg: kernel logs
      - “kernel ring buffer”
- Kernels and Drivers
  - /boot
    - The /boot directory contains the kernel, its associated files, and its needed drivers to boot up the operating system during the startup process.
- Memory Information: ps, top, free
  - Ps (Process Status) - shows top processes running on the system
  - Top - Provides a real-time, dynamic view of system processes.
  - Free - show memory usage
- Optical Media
  - ISO File System (isofs) is the file system for optical disc media

---

#### 4.4 Your Computer on the Network

- Networking Basics:
  - LAN: Local Area Network connects devices within a small area.
  - DNS: Resolves domain names to IP addresses.
  - IPv4 and IPv6: Internet protocols for addressing devices.
  - DHCP: automatic IP addressing
    - Manual, Automatic, and Dynamic are the 3 mechanisms
  - TCP/IP
    - Set of networking protocols that allows two or more computers to communicate

- Networking Commands:
    - ping: Checks connectivity to another device.
    - netstat/ss: Shows network connections and ports.
    - ifconfig/ip addr show: Displays network interface configurations.
      - lo is the loopback interface and is always present
    - Config files: /etc/resolv.conf (DNS), /etc/hosts.
    - Route: used to show/manipulate the IP routing table
- 

## Topic 5: Security and File Permissions

---

### 5.1 Basic Security and Identifying User Types

- User Types:
    - Root User:
      - The superuser with unrestricted access to the system.
      - Used sparingly for administrative tasks.
      - Represented by the GID of 0
    - Standard Users:
      - Regular users with limited access to system resources.
      - Begin at UID 1000 and increment by 1 for each additional user.
    - System Users:
      - Non-login accounts used by system services.
  - Key Files:
    - /etc/passwd: Stores user account information.
      - “Defines all users on a linux system”
    - /etc/shadow: Stores encrypted user passwords.
      - The last 6 fields provide password aging and account lockout features
    - /etc/group: Lists group memberships.
  - Commands:
    - id: Displays user ID, group ID, and group memberships.
    - last: Shows the last login of users.
      - root tty2 Wed Sep 29 21:11 - 21:11 (00:00)
    - who: Displays currently logged-in users.
    - w: displays more info about logged on users and what they are doing
  - Privilege Elevation:
    - sudo: Runs commands as another user (typically root).
    - su: Switches to another user account.
- 

### 5.2 Creating Users and Groups

- User and Group Management:
  - Users and groups allow controlled access to system resources.

- Key Files:
    - /etc/passwd: Stores user account details.
    - /etc/shadow: Contains password and expiration information.
    - /etc/group: Lists group memberships.
    - /etc/skel/: Template directory for creating new user home directories.
  - Commands:
    - useradd: Creates a new user.
      - -d to specify the users home directory location instead of using the default
      - -m to create a user without a home directory
    - groupadd: Creates a new group.
    - passwd: Sets or changes user passwords.
    - chgrp to change group ownership of a file
    - chown to change the owner of a file
    - chmod to change permissions of UGO to a file
- 

### 5.3 Managing File Permissions and Ownership

- File Permissions:
    - Permissions determine who can read, write, or execute a file.
    - Represented as rwx (read, write, execute).
      - Example: -rwxr-xr--
        - Owner: rwx (read, write, execute).
        - Group: r-x (read, execute).
        - Others: r-- (read only).
      - Rwt
        - Indicates that the directory is globally writable, but only the owner can delete their own files within the directory
    - Use umask to change the default permissions for files and directories at the time of creation
  - Ownership:
    - Each file has an owner and group.
    - Commands:
      - ls -l: Displays file permissions and ownership.
      - chmod: Changes file permissions.
      - chown: Changes file ownership.
- 

### 5.4 Special Directories and Files

- Special Directories:
  - /tmp/ and /var/tmp/:
    - Temporary directories for storing files.
    - /tmp/ is cleared upon reboot, while /var/tmp/ persists across reboots.

- Special Permissions:
    - Sticky Bit:
      - Applied to directories to restrict file deletion to the owner.
      - Example: `chmod +t /tmp/`.
  - Symbolic Links:
    - Shortcuts or pointers to other files or directories.
    - Command: `ln -s target link_name.` file to the screen at once
- 

### 1st Attempt Notes

- To remove a directory
  - `rm -r`
  - `rmdir`
- Study tar a little bit more
  - Create a compressed archive
    - `tar -cf archive_name.tar files_to_include`
  - Extract an archive:
    - `tar -xf archive.tar`
- Where does Linux store its documentation, usage, and example files?
  - `/usr/share/doc`
- How to get more info on how to use a command in Linux?
  - `info command`
  - `command --help`
  - `man command`
  - `apropos command`
- If someone has experience with Red hat linux enterprise but does not want to pay for the license, what is the closest free alternative to it?
  - CentOS
  - Fedora
  - Oracle Linux
  - Debian
- What is the default package manager on RHEL?
  - `yum/dnf`
  - `Rpm` (Low Level)
- Root User is what UID?
  - `0`
- Understanding the `rwX` permissions
  - First letter
    - `D` for directory
    - `-` for file
  - Last letter

- T instead of X means it's sticky
    - Only the owner of the directory can delete or modify files within the directory
- What is used to resolve DNS queries
  - Host
- Where do you find config files?
  - /etc
- What is dmesg
  - Shows kernel log messages
    - A command on most Unix-like OS that prints the message buffer of the kernel
    - Contains messages produced by the device drivers and serves to show the kernel's log messages to the screen
- Understand common Linux Apps
  - Thunderbird
    - Mail client
  - Apache
    - Web server
  - GIMP
    - Image manipulation
  - Postfix
    - Mail Server
      - Routes and delivers electronic mail
- How do you split long commands in the Linux terminal?
  - \
- Two extensions to use tar on?
  - bz2 + gz
-