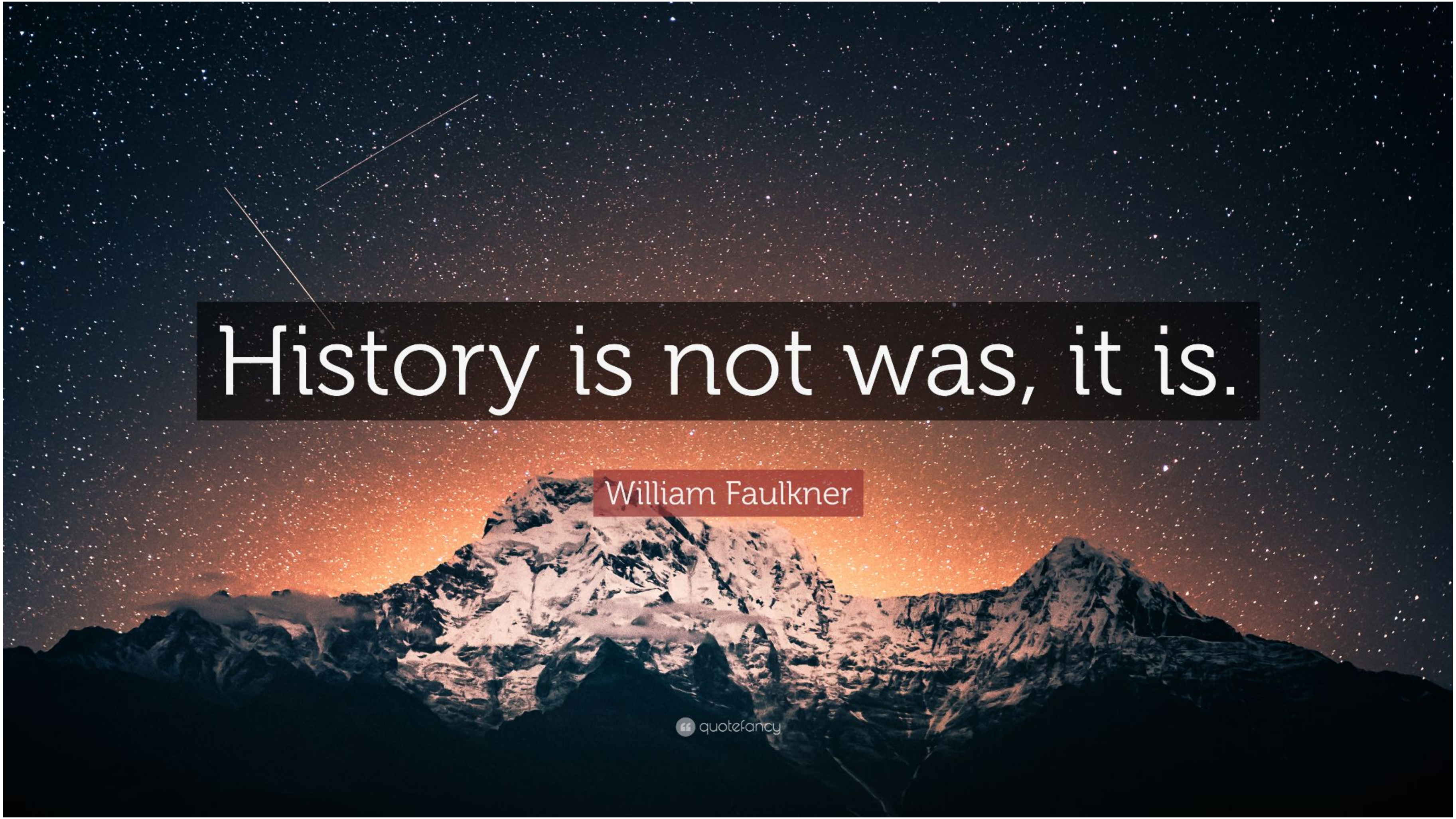
Unix Command Line

Kameswari Chebrolu
Department of CSE, IIT Bombay

https://pbs.twimg.com/media/E-YJGozUUAA6rUU.jpg

Outline

- Unix history and why popular?
- Command line vs GUI
- What is a Shell?
- Linux File System
- Various commands



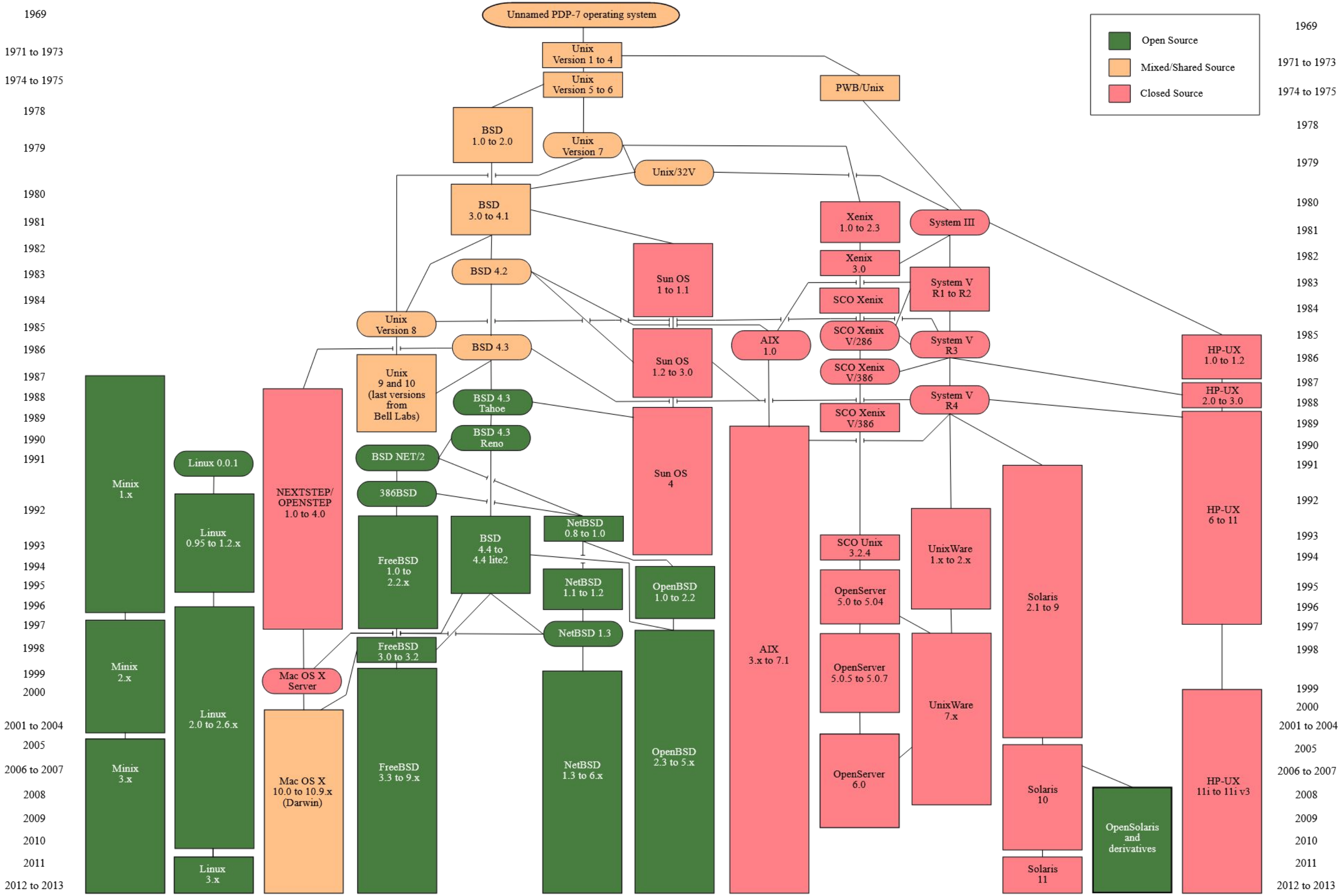
History is not was, it is.

William Faulkner

Unix/Linux OS

- Unix: Proprietary OS created in late 1960s at AT&T Bell Labs
- Linux: a clone of Unix, free and open source
  - Written from scratch by Linus Torvalds in 1991

- Distributions of Linux: Linux OS packaged with lot of additional free software
  - Fedora, Ubuntu, CentOS, SuSe etc
  - Differ wrt to desktop environment, package installation, display server etc
  - Other Unix clones: FreeBSD and Mac OS X (its kernel Darwin, is based on BSD)
- A user on one Unix system can move to another easily wrt to command-line



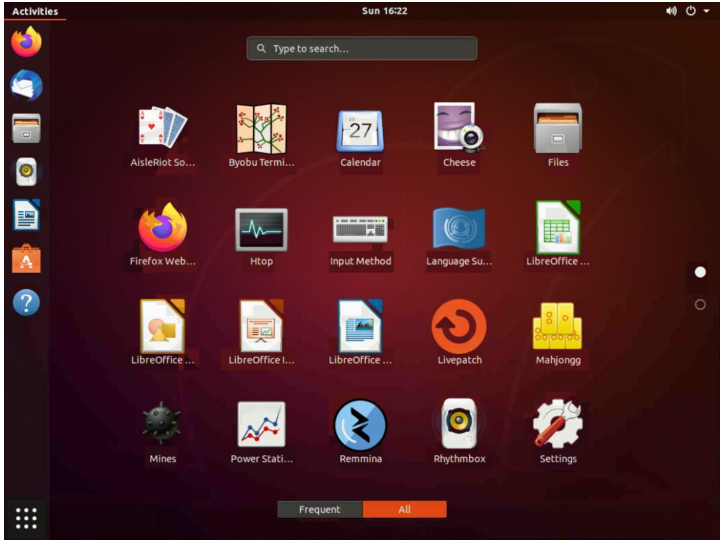
https://sosheskaz.github.io/technology/2017/05/12/Adventures-In-Bsd.html

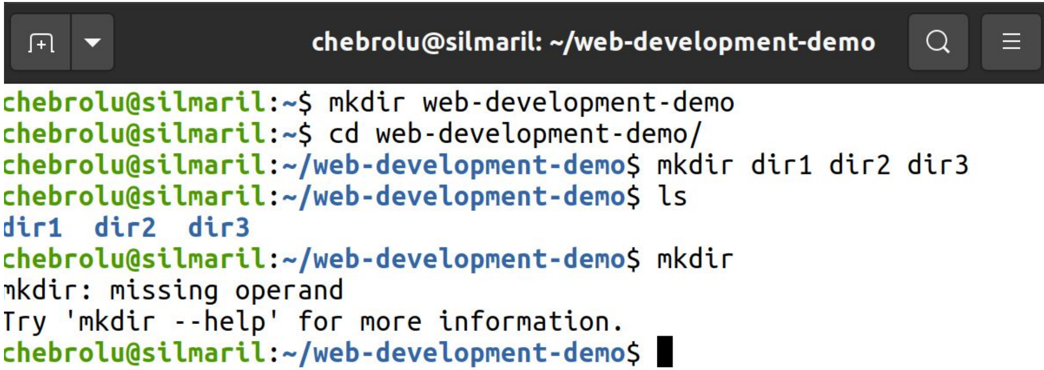
Popularity of *nix

- “Since we are programmers, we naturally designed the system to make it easy to write, test, and run programs” – Unix Creators, Dennis M. Ritchie and Ken Thompson
  - Very server and programmer-friendly OS
  - Linux (FREE) is for developers!
  - Easy to do scripting
  - Lot of scientific libraries and programs are written for *nix

- Open source (some versions) and exposes you to an ecosystem of open-source software
  - Helps bridge the concepts you learn with how they're applied in practice.
    - Interested in OS? Dig into details of open source linux and interaction with device drivers
    - Interested in Compilers? Clone gcc source
    - Interested in distributed systems? Clone Hadoop and run a cluster on your laptop
    - Interested in cloud computing? Containers origins in linux

Command Line vs GUI





```
chebrolu@silmaril:~$ mkdir web-development-demo
chebrolu@silmaril:~$ cd web-development-demo/
chebrolu@silmaril:~/web-development-demo$ mkdir dir1 dir2 dir3
chebrolu@silmaril:~/web-development-demo$ ls
dir1  dir2  dir3
chebrolu@silmaril:~/web-development-demo$ mkdir
mkdir: missing operand
Try 'mkdir --help' for more information.
chebrolu@silmaril:~/web-development-demo$ 
```

Windows GUI: use pre-programmed interface ⇒ set of possible actions pre-decided

Command-line Shell: a prog. (scripting) language ⇒ use pre-written programs **AND** compose new scripts!

Power of the Shell

Alias: shell, terminal, console, prompt etc

1. Rename a set of files
2. Number of lines in all C files in a directory
3. Top five files with maximum number of lines

Demo!

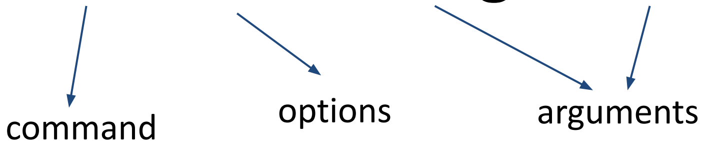
A Brief History of the Shell

- Unix: OS for mainframe computers
  - Users connecting remotely via individual terminals (keyboard and screen)
  - No local programs, send text and receive text
  - Terminals based on text since text is light on resources
  - Commands kept very terse to reduce the number of keystrokes needed

- Need to support all kinds of file management tasks
  - Create files, list files, rename, move to folders etc
  - Each task required its own program (or command)
  - **Master program to coordinate execution of all these programs → shell**
- Original Unix shell called sh (Bourne shell)
  - Extended with better features and syntax is BASH (Bourne Again SHell)
  - Other shells also: zsh (mac OS), csh, fish etc

Basic Instructions

- Open shell: Click on “Activities” top left of the screen + type shell in the search box (or) use Ctrl-Alt-T
- Type a command in the same line as where $ (prompt) appears (command line ;-)
- Commands sometimes have number of arguments (command-line arguments)
  - tar -zcvf lab1.tgz lab1/



command

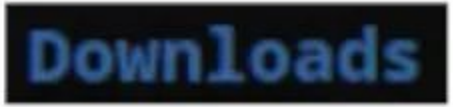
options

arguments

- The shell does not execute commands until the “Enter key” is pressed
- Any output the shell produces will usually be printed directly in the terminal
  - Another prompt is shown once finished
- Commands are case sensitive (ls vs LS)

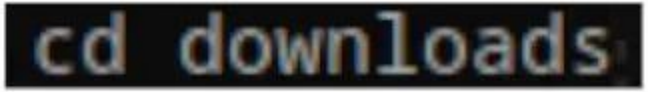
Demo!

my folder :



```
Downloads
```

me :



```
cd downloads
```

Linux :

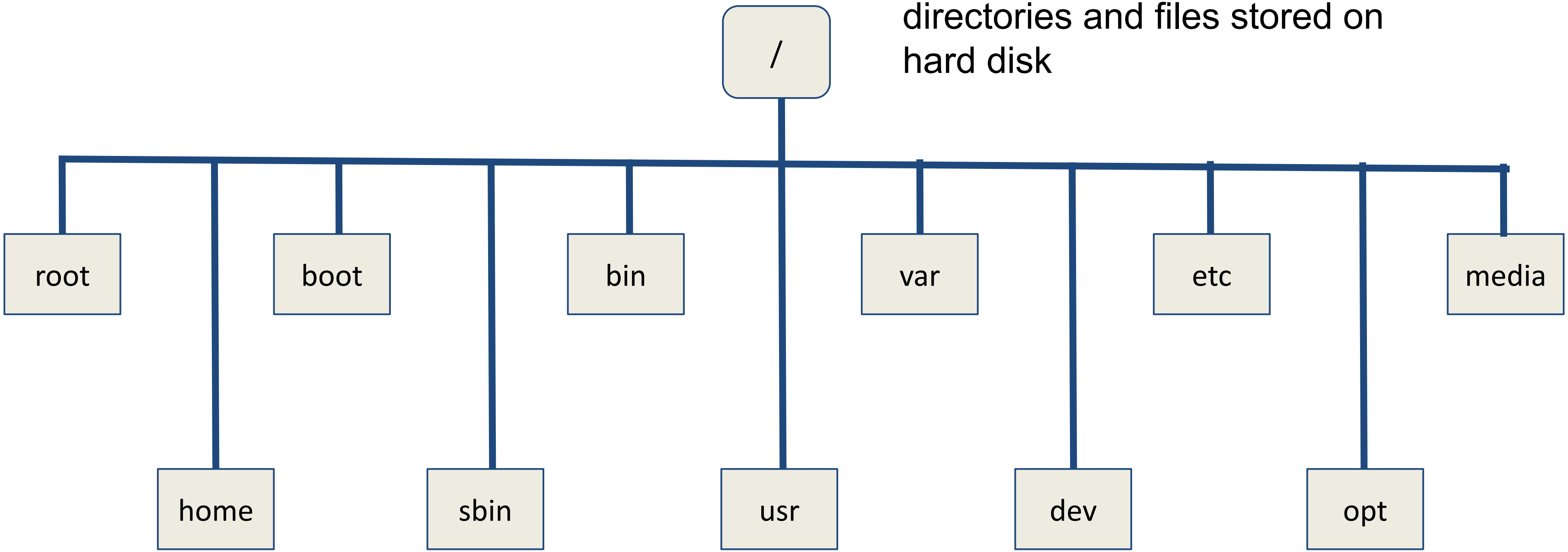
https://preview.redd.it/3zoe90gmq7y51.jpg?auto=webp&s=791da8fa068769a9e73f494f63e730c7f354fb86

Outline

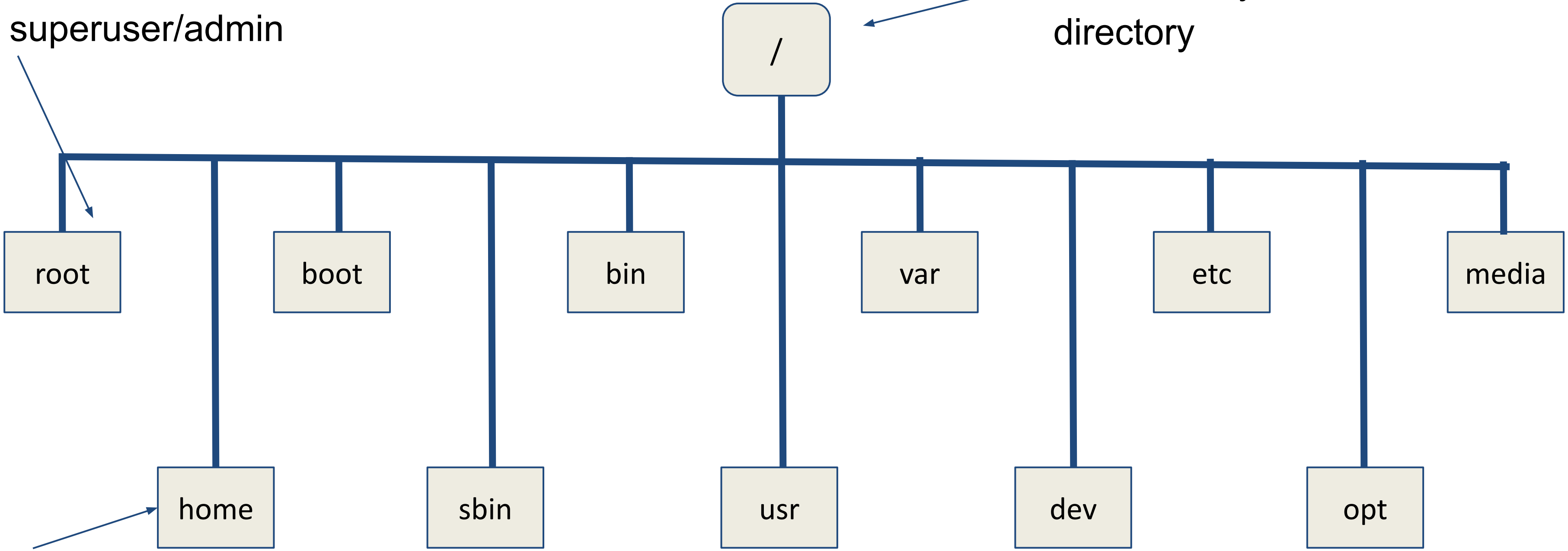
- ~~Unix history and why popular?~~
- ~~Command line vs GUI~~
- ~~What is a Shell?~~
- Linux File System
- Various commands

Linux Filesystem

A filesystem is a hierarchy of directories and files stored on hard disk



Linux Filesystem



The file system starts with the directory called the root directory

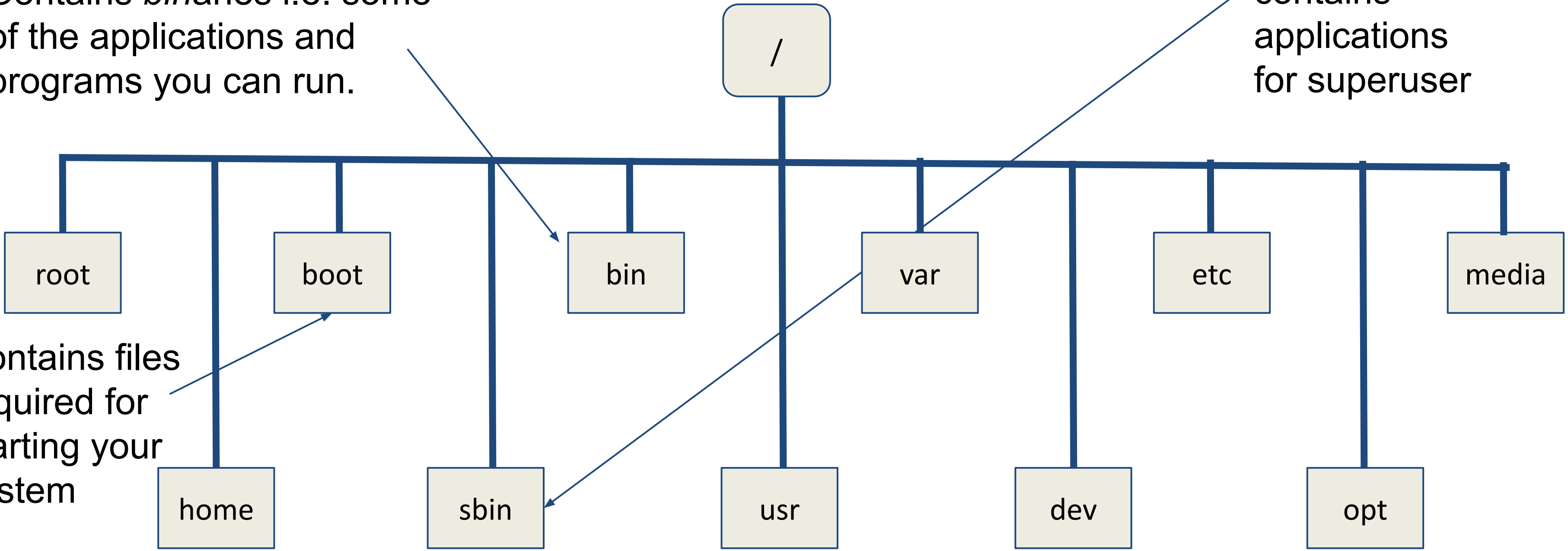
Home directory of superuser/admin

User's personal directories

Linux Filesystem

Contains *bin*aries i.e. some of the applications and programs you can run.

sbin is similar to /bin, but contains applications for superuser

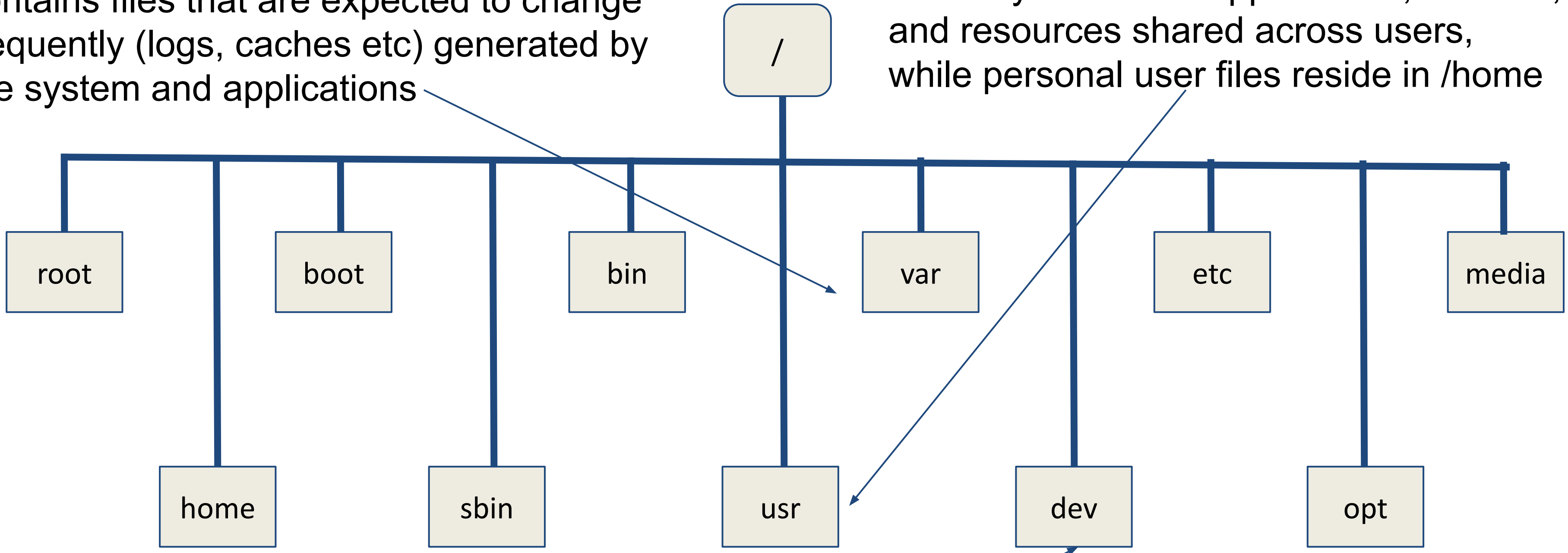


Contains files required for starting your system

Linux Filesystem

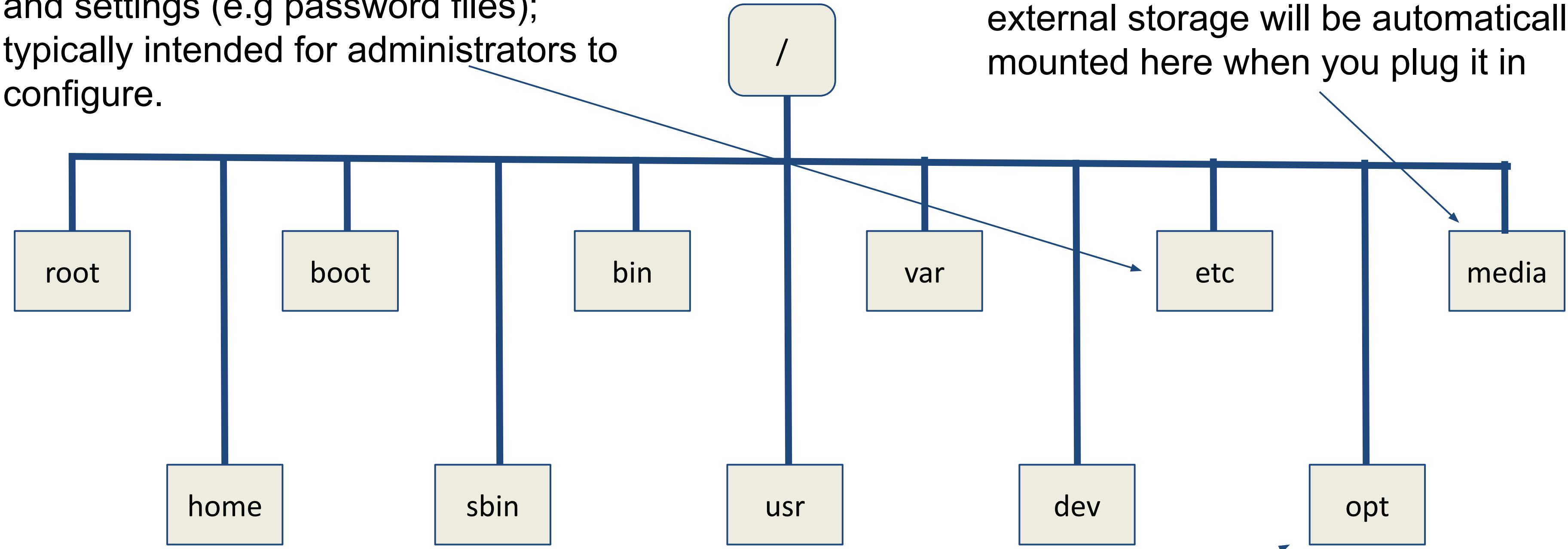
contains files that are expected to change frequently (logs, caches etc) generated by the system and applications

Holds system-wide applications, libraries, and resources shared across users, while personal user files reside in /home



Contains device files (hard drives, USB devices, webcams etc). These files allow software to interact with hardware as if they were standard files

Linux Filesystem



holds system-wide configuration files and settings (e.g password files); typically intended for administrators to configure.

external storage will be automatically mounted here when you plug it in

optional or third-party software packages that aren’t part of the default Linux distribution

File and Directory Commands

- clear
- man
- pwd
- ls
- cd
- mkdir
- rmdir
- cp
- mv
- rm

- These commands enable users to
  - Navigate the file system
  - Create, move or remove files and directories
- Provide a powerful interface for interacting with the operating system

Clear

- Clears the terminal screen
  - Terminal cursor moves to the top-left corner
- Helps enhance readability
  - Use before running new commands to avoid clutter and improve focus
- Note: Doesn't delete history or affect running programs
  - Only affects visual display

man

- Displays manual (help) pages for Unix commands
- Useful when learning new commands or options you are unfamiliar with
- Provides detailed documentation
  - Descriptions, options, usage examples, and technical details

  - Usually formatted with a consistent structure
    - NAME, SYNOPSIS, DESCRIPTION, OPTIONS, EXAMPLES, SEE ALSO.
  - Use the arrow keys or page up/down to scroll through the manual
  - Press / followed by a keyword to search within the manual page
  - Press q to quit the manual page
- Syntax: man [command]

pwd

- Shell has a notion of a default location
  - For the root user, home is at /root
  - Regular users, it is /home/username (e.g. /home/chebrolu)
- pwd (present working directory) command tells your current working directory
  - No options needed
  - Displays the full path of the directory you are currently in

- Use Case:
  - Helpful when navigating directories
    - Use pwd to confirm your current directory, especially when working in deep or complex directory structures
  - Helpful with scripting and automation
    - Dynamically get the current directory and perform operations relative to it

Demo

man, clear, pwd

ls

- ls: display contents of the current directory
  - Directories often listed in a different color (e.g., blue)
  - Executable Files may be displayed in green
- Syntax: ls [Options] [Files/Directories]
- Use Case:
  - Quickly see what files and directories exist in your current or specified directory
  - Checking file details like permissions or file size

- Key Options:
  - -l : Shows detailed information
    - File permissions, number of links, owner, group, size, and modification date
  - -lh : Displays file sizes in a human-readable format (e.g., KB, MB)
  - -lt : Sorts the output by the time of last modification, with the newest files first
  - -a : display all files including the hidden files
    - Every directory has at least two entries: “.” and “..” (called dot and dotdot)
      - dot directory is a shortcut for the current directory
      - dotdot is a shortcut to the parent directory
  - -R: list subdirectories recursively
  - -S: sort by file size, largest first
  - -X : sort alphabetically by entry extension

cd

- Changes the current working directory
  - Absolute paths:
    - “/” at the start of your path means “starting from the root directory”
    - (”~”) at the start of your path means “starting from my home directory”
  - Relative Path: Starts from the current directory
    - e.g. ../folder (moves up one directory)
- Syntax: cd [directory]
  - Directory you want to navigate to
  - If omitted, cd defaults to the home directory
- Use Case: efficient file system navigation
  - Enables users to work effectively within different directories

- Key Options:
  - No Options: takes you to your home directory
  - .. : Moves you up one directory level
  - - : Switches to the previous directory
  - ~ : Represents home directory, useful for quickly navigating there
- “Tab” for auto filling
  - Applies to all commands, not just cd!

Demo

ls, cd