

Unix Command Line

Kameswari Chebrolu

Windows users



Have patience

Mac users



Have money

Linux users



Have skills

Outline

- Unix history, why Unix?
- Command line vs GUI
- What is a Shell?
- Linux File System
- Various commands (basic)



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

Why *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



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

```
mark@linux-desktop: /tmp/tutorial

File Edit View Search Terminal Help

mark@linux-desktop:~$ mkdir /tmp/tutorial
mark@linux-desktop:~$ cd /tmp/tutorial
mark@linux-desktop:/tmp/tutorial$ mkdir dir1 dir2 dir3
mark@linux-desktop:/tmp/tutorial$ mkdir
mkdir: missing operand
Try 'mkdir --help' for more information.
mark@linux-desktop:/tmp/tutorial$ cd /etc ~/Desktop
bash: cd: too many arguments
mark@linux-desktop:/tmp/tutorial$ ls
dir1 dir2 dir3
mark@linux-desktop:/tmp/tutorial$
```

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

Power of the Shell

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

A brief history

- Alias: shell, terminal, console, prompt etc
- 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/



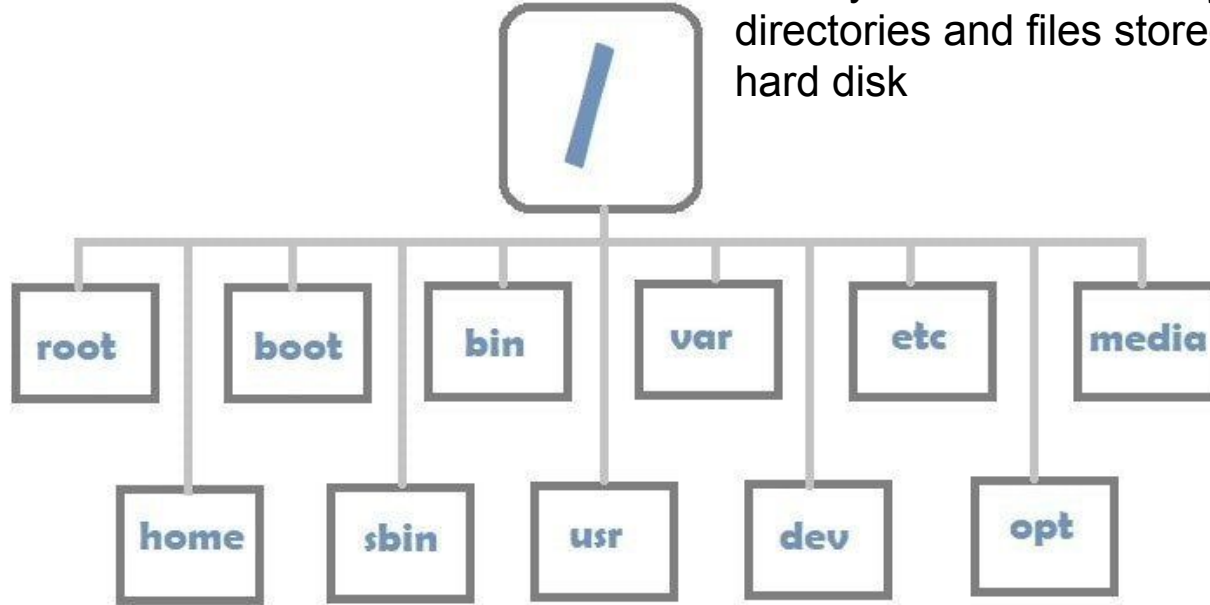
Diagram illustrating the components of the command `tar -zcvf lab1.tgz lab1/`:

- `tar` is labeled as the **command**.
- `-zcvf` is labeled as the **options**.
- `lab1.tgz` and `lab1/` are labeled as the **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)

Linux Filesystem

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

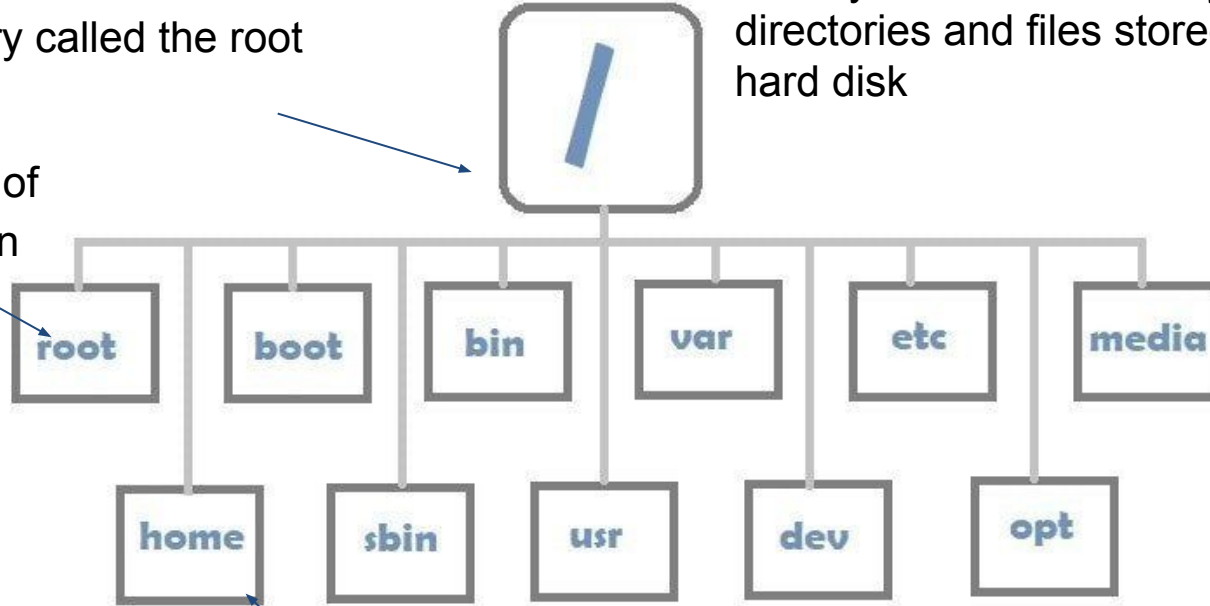


Linux Filesystem

The file system starts with the directory called the root directory

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

Home directory of superuser/admin



User's personal directories

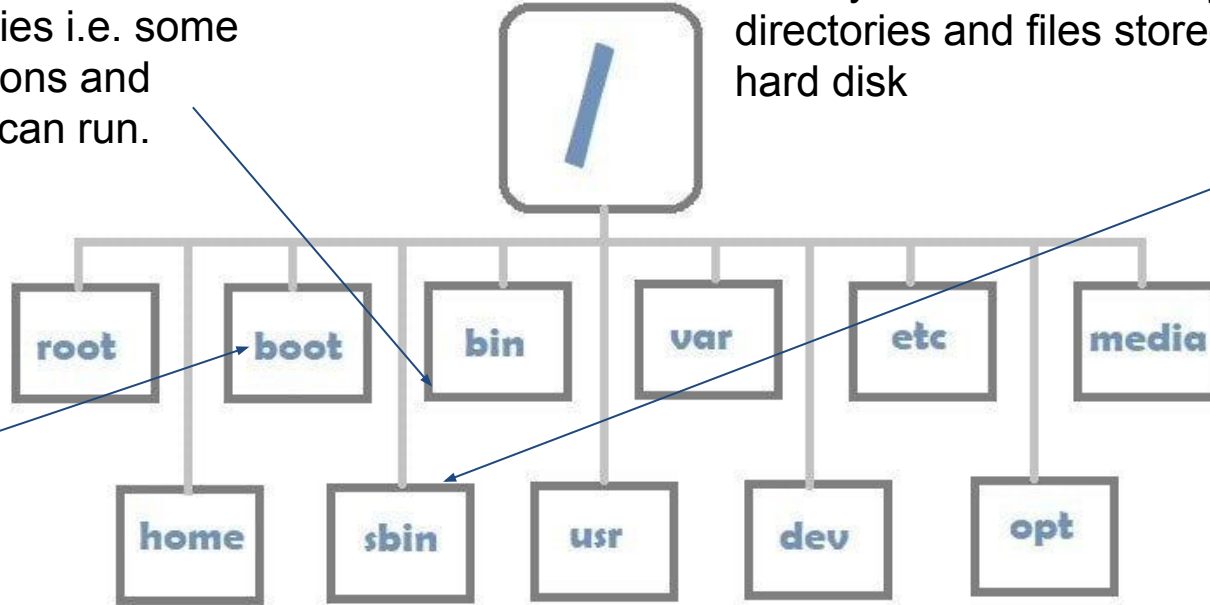
Linux Filesystem

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

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

Contains files required for starting your system

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



Navigate file system: pwd and cd

- 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/kc)
- Pwd (present working directory) command tells your current working directory

- You can change the working directory using “cd” command
 - Relative and absolute paths
 - 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”
 - “Tab” for autofilling
 - Applies to all commands, not just cd!

- ls: display contents of the current directory
 - Syntax: ls [OPTIONS] [FILES]
 - ls -l (print files in a long listing format)
 - Short cut ll
 - ls -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
 - Checkout options: -X, -S, -t, -R
- Lets explore through examples

Create folders/files: mkdir, echo, cat, ls

- mkdir (make directory): create a directory
 - Syntax: mkdir [OPTION] [DIRECTORY]
 - Takes one or more directory names as its arguments.
 - “-p” option: creates the directory only if it doesn’t exist.
- echo: create and populate files via redirection (>)
 - Displays text on the screen
 - Handy when writing scripts

- Cat (concatenation): display the contents of a text file to screen
 - Why named cat then? can combine outputs also
 - “-n” option to display contents of a file with line numbers
 - “-s” option to omit the repeated empty output lines:
 - Big file: use “less” command
 - Can use Up Arrow, Down Arrow, Page Up, Page Down, Home and End keys to move through your file
- Lets explore some examples

Manipulating files/folders: mv, cp, rm, rmdir

- mv: move files or folders
 - Syntax: mv [OPTIONS] SOURCE DESTINATION
 - SOURCE can be one, or more files or directories, and DESTINATION can be a single file or directory
 - When multiple files or directories are given as a SOURCE, the DESTINATION must be a directory
 - In this case, the SOURCE files are moved to the target directory.
 - If you specify a single file as SOURCE, and the DESTINATION target is an existing directory, then the file is moved to the specified directory.
 - If you specify a single file as SOURCE, and a single file as DESTINATION target then you're renaming the file .
 - When the SOURCE is a directory and DESTINATION doesn't exist, SOURCE will be renamed to DESTINATION. Otherwise if DESTINATION exist, it be moved inside the DESTINATION directory.
 - Use wildcards for ease of moving
 - Checkout options: -i, -n, -f

- **cp:** copy files or folders
 - Very similar to mv, except it copies files instead of moving
 - Checkput options: -R, -i, -p
- **rm:** remove files
 - “-i” option: prompt before removing
 - “-r” recursively remove and also directories via -r
- **rmdir:** remove folder (but folder has to be empty)
 - rm -d is same as this

- **Caution:** rm doesn't move files to a folder called "trash" or similar
 - Deletes them totally, utterly and irrevocably
 - Be extra careful when using wildcards!!!
 - Use -i (interactive) option to rm, which will prompt you for confirmation
 - Y to delete it, N to keep it, and Ctrl-C to stop the operation
- Lets see some examples!

Miscellaneous

- man: manual for commands
 - E.g. man ls
- clear: helps clear the screen to reduce clutter
- history: shows history of all commands you entered
 - At terminal use !number to run that command in history
 - !! runs previous command
 - history 5 (shows last 5 commands)
- du: disk usage, used to estimate file space usage
 - Checkout options: -h, -a, -s
- head/tail: print n lines from head or tail
 - head file; head -n 2 file
 - Checkout: -n and -c option
- which: identify the location of a given executable(s)

(Simple) Editors

- Vi
- Nano
- gedit

(write some text, delete some text, save, more editing, save and exit, reopen)

References

- <https://ubuntu.com/tutorials/command-line-for-beginners#1-overview>
- <https://www.linuxfordevices.com/tutorials/linux/nano-editor-in-linux>
- <https://www.linuxfordevices.com/tutorials/linux/vim-tutorial>

Misc.

- File System:

<https://www.linuxfoundation.org/blog/blog/classic-sysadmin-the-linux-filesystem-explained>

- Figure of Unix variants:

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