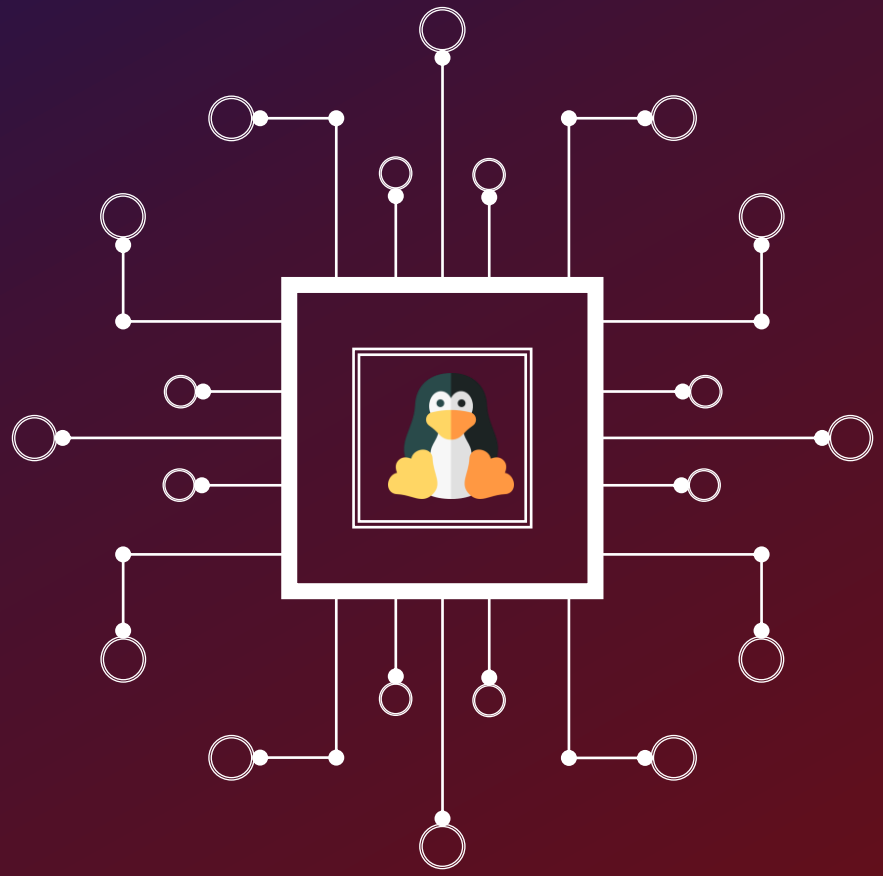
Using Shell

All the best people in life seem to like LINUX.
Steve Wozniak



Making Linux GPL'd was definitely the best thing I ever did.
Linus Torvalds

Amirkabir Linux Festival 2021

Preface

- Before icons and windows took over computer screens, you typed commands to interact with computers.
- The program used to interpret and manage commands was referred to as the shell.

The following are a few major reasons to learn how to use the shell:

- You will learn to get around any Linux or other UNIX-like system.
- Special shell features enable you to gather data input and direct data output between commands and Linux filesystems. To save on typing, you can find, edit, and repeat commands from your shell history.
- You can gather commands into a file using programming constructs such as conditional tests, loops, and case statements to perform complex operations quickly, which would be difficult to retype over and over.

What's a Command?

- Typically a program name followed by options and arguments

wc -l myfile

ls -l -a –t **ls -lat**

- Options, which usually begin with a dash, affect the behavior of the program

- Test these:
  - date
  - pwd
  - hostname
  - ls

Commands with Examples

- Listing files (ls)
  - ls
  - ls -R
  - ls -a
  - ls -al
- Viewing (cat)
  - cat a
  - cat a b > c
- history
- clear
- echo
- copy / paste

Getting Help

- man
  - man man
  - man ls
  - man wc
- info
  - info info
  - info ls
  - info wc
- ls --help

Sudo

- Linux is a multiuser operating system
- “sudo” was developed as a way to temporarily grant a user administrative rights

Locating commands

- where these commands are located and how the shell finds the commands you type?
  - type
  - which

Connecting Commands

- Piping between commands (|): connects the output from one command to the input of another command
  - cat /etc/passwd | sort | less
- Sequential commands (;): sequence of commands to run, with one command completing before the next command begins
  - ls ; date
- Background commands (&): you may not want to tie up your shell waiting for a command to finish
- less than (<) and greater than (>) signs

Variables

- shell itself stores information that may be useful to the user's shell session in what are called "variables"
  - $SHELL
  - $USER
- Environment variables are variables that are exported to any new shells opened from the current shell
  - env
- Set an Unset Local variables
  - Variable_name=value
  - unset Variable_name

Locating commands

- To find commands you type, the shell looks in what is referred to as your *path*
- The path consists of a list of directories that are checked sequentially for the commands you enter
  - Echo $PATH

Creating and using aliases

- you can effectively create a shortcut to any command and options that you want to run later
    - alias p='pwd ; ls'

Configuring your shell

- rc file: contains the information that is specific to your bash shells
- This is the best location to add aliases so that your shell picks them up
    - ~/.bashrc

Basic Commands You Should Know

- ls
- mkdir
- rm
- more / cat
- free
- cp / mv
- touch
- head / tail
- cd
- bg / fg
- top
- kill
- whoami
- free
- df / du
- ifconfig
- wget
- Ctrl + C / Z
- history
- man
- echo
- TAB
- CTRL + A / E
- hostname
- CTRL + W / K
- zip
- history

More Information

- The Linux Command Line: A Complete Introduction, William E. Shotts Jr.
- Linux bible, Christopher Negus. Chapter 3. Using the shell
- Linux Pocket Guide