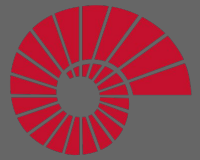
Introduction to Linux Shell



KOÇ UNIVERSITY

COMP201 Lab Session
Fall 2020

Agenda

- Linux
- Options to Install a Distro
- Ubuntu Shell Introduction
- Basic Shell Commands

What is Linux?

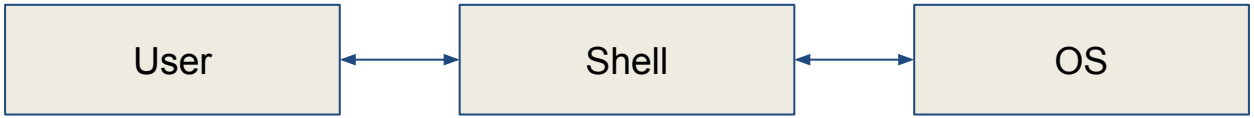
- Linux is a free and open-source operating system (OS) kernel.
- Combined with a GNU system, Linux forms an OS.
  - https://www.gnu.org/gnu/linux-and-gnu.en.html
- There are many GNU/ Linux distributions like Ubuntu, Fedora, Arch Linux, OpenSUSE, etc.
  - You may install Ubuntu 18.04 Desktop for this course.
    - https://www.ubuntu.com/

Options to Install a GNU/Linux Distro

- You can install a distro using:
  1. Virtualization using Oracle VirtualBox, WMware, etc.
     - https://www.virtualbox.org/
  2. Dual-boot (multi-boot)
  3. Format (installing it as main OS)

What is Shell?

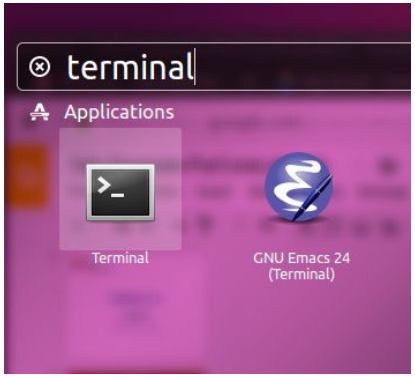
- Shell is the interface between the user and the OS
  - Takes command from user or script and gives them to OS
- Shell is also often called Terminal
- There are many applications for terminal:
  - **Bash**, Sh, Zsh, Csh, etc.



Terminal

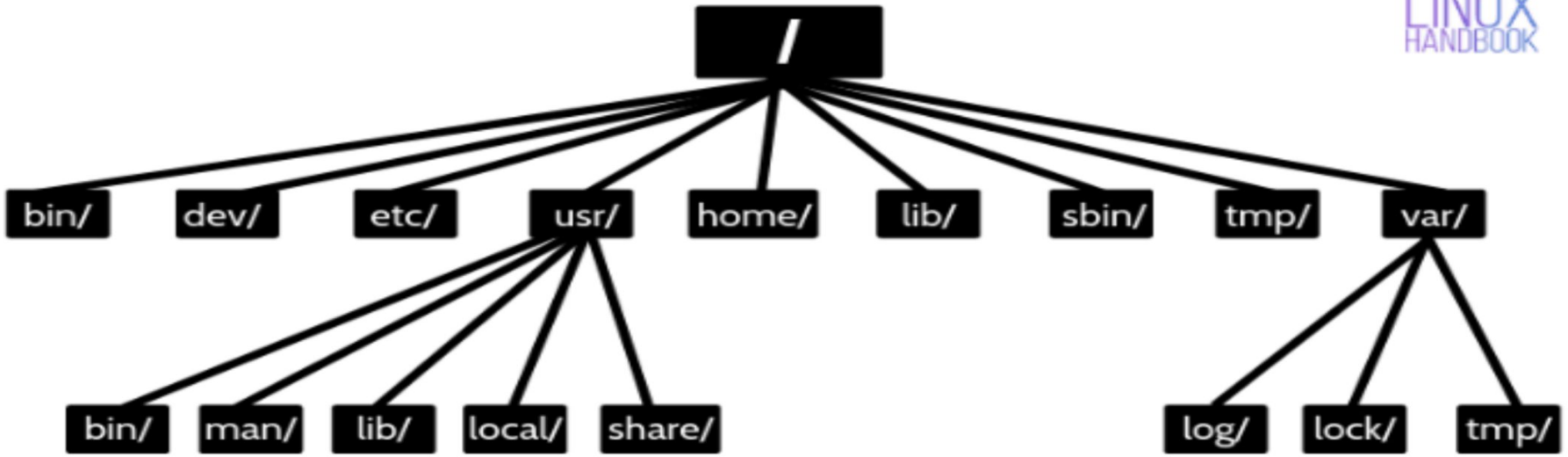
- Accessing terminal
  - Search terminal using Ubuntu search button

Shell is a program which processes commands and returns output , like bash in Linux . **Terminal is** a program that run a **shell**



Directory Hierarchy

- Linux has hierarchical directory structure
  - Files organized as tree-like pattern of directories



Basic Navigation Commands

- Current directory is called working directory
  - `pwd` command: Print working directory
  - `echo` command: Print the input passed to it
  - `ls` command: List files in working directory
  - `cd` command: Change working directory
    - Dot operator (.): Current directory
    - Double dot operator (..): Working directory's parent
    - ~ operator: Change to home directory

Creating Directories/Files

- Creating directory
  - `mkdir` *directoryname*
- Create a file with desired name and extension
  - `Touch` *filename.ext*
- Output a file's content
  - `cat` *filename.ext*
- Clear terminal output
  - reset
  - clear

Deleting Directories/Files

- Deleting a file
  - `rm` *filename*
  - rm -rf directory (force delete recursively)
- Deleting a directory
  - `rmdir` *directory*
  - *There should be no files inside*

Copying/Renaming Files

- Copy file
  - `cp` *filename1 filename2*
- Rename/move file
  - `mv` *oldfilename newfilename*

Manipulating Files

- Searching file contents
  - Using `less` command
  - Using `grep` command
- Word count
  - `wc` *filename*

  Use `-c` for characters, `-w` for words, -l lines

I/O Redirection

- Redirecting output
  - > operator (write output in file)

```
ls > list.txt
```

- Append output
  - >> operator (append output in file)

```
Ls >> list.txt
```

- Redirecting input
  - < operator (read input from file)

```
sort < list.txt
```

Pipes

- Redirect output of one program to input of another program

```
ls | wc
```

- You can pipe multiple programs to each other, output of each one becomes the input of the next one:
  cat list.txt | sort | uniq | wc

Environment Variables?

- A set of variables sent by the calling process (e.g., shell) to the new program being executed
- Some standard bash variables

–$PATH: Colon separated directory list for command search

–$HOME: Currents user's home directory

–$LOGNAME: Current user's name

–$SHELL: User's preferred shell

–$EDITOR: User's preferred editor

$PATH

- For example:

  /bin:/usr/bin:/sbin

- When you enter a command in bash (e.g., ls), it will search for an executable file with that name in directories listed in $PATH one by one, and executes the first one found
- Hence, to run an executable in current directory, we need to use ./ before the program name

Installing Compiler

- Install the package *build-essential* to have compiler and build tools:

```
sudo apt-get install build-essential
```

- Gcc and G++ and make included ☺
- Some other useful tools to install:

```
sudo apt-get install htop vim emacs git
```