Linux Operating Systems & Command-line interface

Lecture 9

Mr. Uttam Acharya

Learning Outcomes

By the end of this lecture you will:

- Understand the Linux operating system and its function
- Be familiar with its folder and file structure
- Understand Linux file permissions
- Be familiar with text based command line functions in Linux

History of Linux

- Linux began in **1991** as a personal project by Finnish student Linus Torvalds.
- Since the release of its **open source code**, it has grown from a small number of C files to 23 million lines of code.
  - Started it for fun but ended up with such a large project.
- It is **free** operating system
  - *Some would say:* Limited support (vastly improved – written by contributors)

Linux OS features

- As with file system construction; operating system's all operate in a different manner (ext3, ReiserFS)
- Mostly used in server OS for web servers, database servers, file servers, email servers
- This makes the software non-compatible between systems(Windows Linux)
- It has features like:
  - **Open Source** - availability to all contributors
  - **Multi-User environment** - access many resources disk/ram/application at same time
  - **Multiprogramming** – multiple applications can run at same time
  - **Standard File System** – Hierarchy of directories and files
  - **Security** – authentication protection, controlled access and encryption
  - **Shell** – Command line interpreter

Comparison (Windows vs Linux)

Topic	Linux	Windows
Price (cost)	YES	
Ease of Use (graphic interface)	YES	YES
Reliability (up-time)	YES	
Software (availability)		YES
Hardware (compatibility)		YES
Security (vulnerability)	YES	
Support		YES

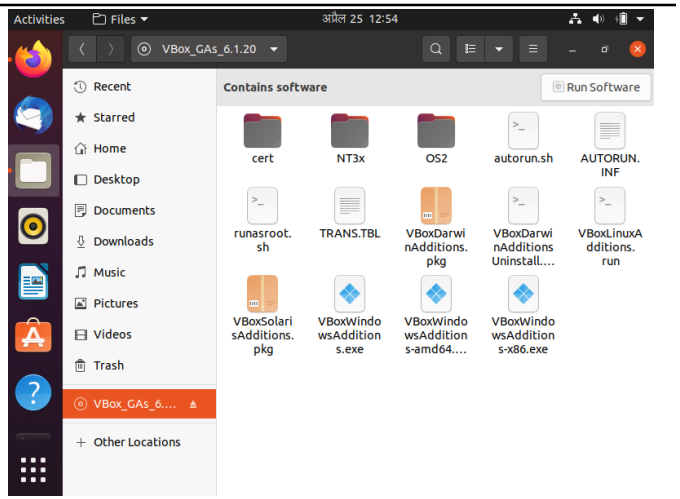
- Result **50/50 depends on user requirements**

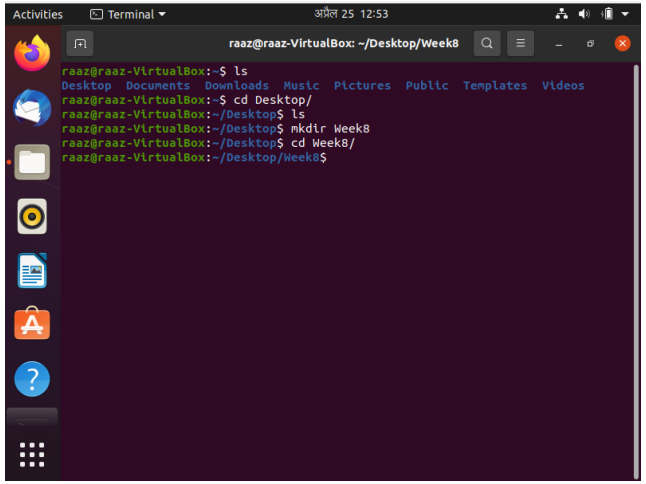
Using the OS (Linux)

Very (*very*) similar to the Microsoft Windows **User Interface** –

- Graphical
- (**a window** + mouse)
- Linux desktop

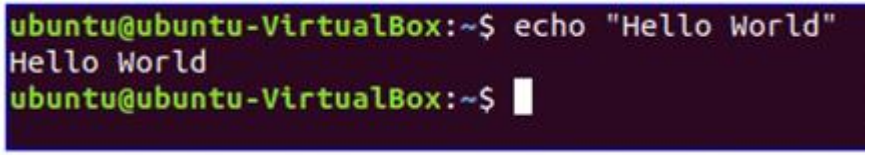
- Text based
- (**command line** + type)
- [**terminal**]





Terminal

- **Terminal** is a command line program which is pre-installed on Linux
- Allows you to navigate through folders and execute files (file and folder management)
- Gives easy access to add **arguments** to executable programs or commands
  - E.g.



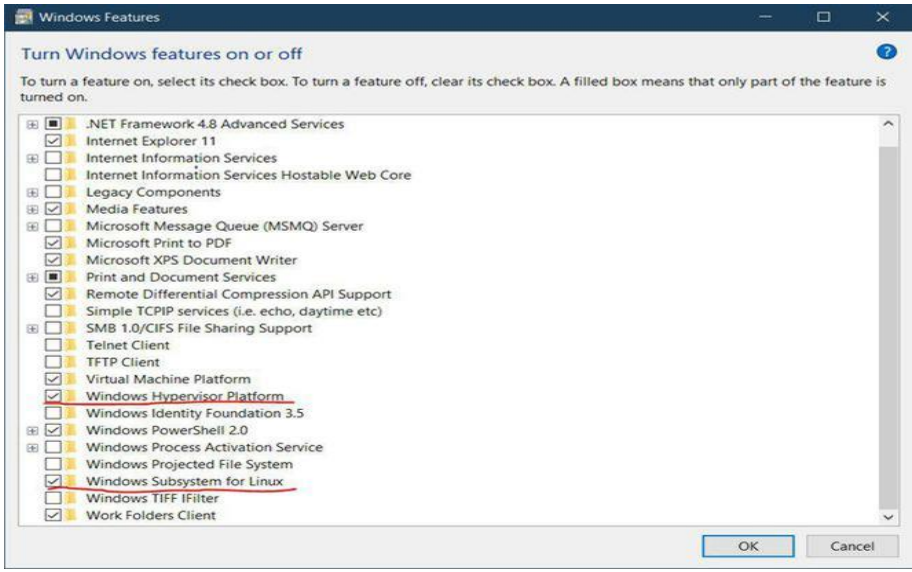
- Windows allows you to double click on executables, some versions of Linux need the **terminal** - ./nameofprogram.sh

How do you Run Linux Terminal on Windows?

Windows Subsystem for Linux (WSL)

Step 1: Enable the Windows Subsystem for Linux optional feature.

You can enable it using the '**programs and features**' settings.



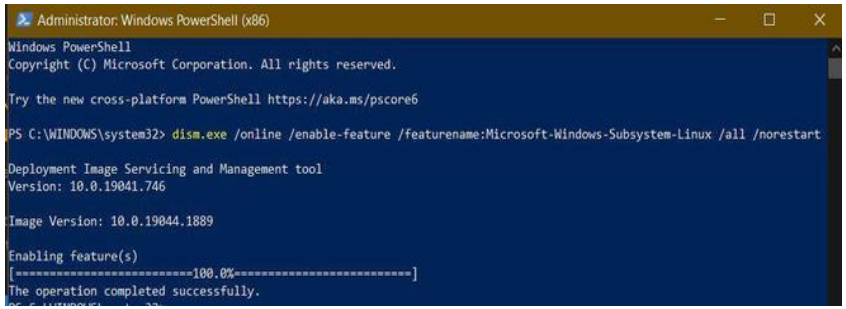
How do you Run Linux Terminal on Windows? (contd..)

Windows Subsystem for Linux (WSL)

Step 2: Enable the Virtual Machine platform and Install WSL2.

The virtual machine has to be enabled before installing WSL, this can be done using the following command.

```
dism.exe /online /enable-feature /featurename:Microsoft-Windows-Subsystem-Linux /all /norestart
```



How do you Run Linux Terminal on Windows? (contd..)

Windows Subsystem for Linux (WSL)

Open Powershell or command-prompt and write the following command.

wsl --install
wsl --set-default-version 2

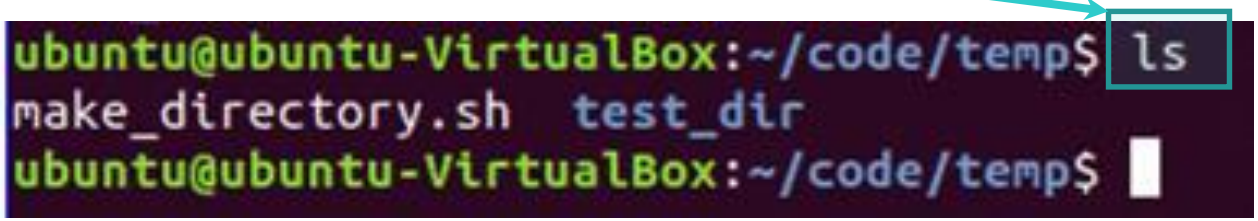
Step 3: Download and Install a Linux distribution from Microsoft Store.

Terminal Command Groups

- **Terminal Navigation Commands –**
  - These commands help the user navigate the system. No amount of terminal knowledge will help you if you can't change directories or get help on a command you don't remember how to use.
- **File Management Commands –**
  - Most Linux distributions come with a graphical desktop environment — but for complicated tasks, it's often easier and faster to use the command line.
    - *$ tar xvzf file.tar.gz -C /path/to/somedirectory*
- **System Management Commands –**
  - These time-tested commands tend to offer a lot more power in terms of what you can do.

Terminal Navigation Commands (some)

- ls [No arguments needed] – list current directory (folder) content with highlighting (files and folders) e.g.



[NB// blue = folder name]
[white = file name]

- pwd – current directory (absolute address)
- cd "NameOfFolder" – change directory (folder)
- man - Displays a help page (manual), very useful for learning how to use a command.

File Management Commands (some)

- **mkdir** – make directory (folder)
- **rmdir** – remove directory (folder)
- **cat** — When used on a single text file, it will display the contents of that file
- **cp** — Makes a copy of a file (same directory)
- **find** — Searches to find files that match a given set of criteria
- **mv** — Moves a file (source □ target)
- **rename** — Changes the name of a file
- **rm** — Removes a file(s) that match a criteria
- **zip/gzip/tar** — Various formats for compressing and decompressing file archives
- **Chmod** — Changes permissions (file / folder)

Permissions

- By default Linux applies permissions to files (and folders)
- There are 3 types of permissions, read, write and execute
- Numerically values can be represented as:

  0 = No Permission, 1 = Execute, 2 = Write, 4 = Read
- We can use **chmod** to change mode (permissions)

  chmod 400 *file* - Read by user
  chmod 040 *file* - Read by group
  chmod 004 *file* - Read by other (World)

  chmod 200 *file* - Write by user
  chmod 020 *file* - Write by group
  chmod 002 *file* - Write by other (World)

  chmod 100 *file* - execute by user
  chmod 010 *file* - execute by group
  chmod 001 *file* - execute by other (World)

Changing Permissions

- For example : **chmod 700**
- Sum the value (by collective; owner, group, world)

chmod **400** *file* - Read by user
chmod 040 *file* - Read by group
chmod 004 *file* - Read by other (World)

chmod **200** *file* - Write by user
chmod 020 *file* - Write by group
chmod 002 *file* - Write by other (World)

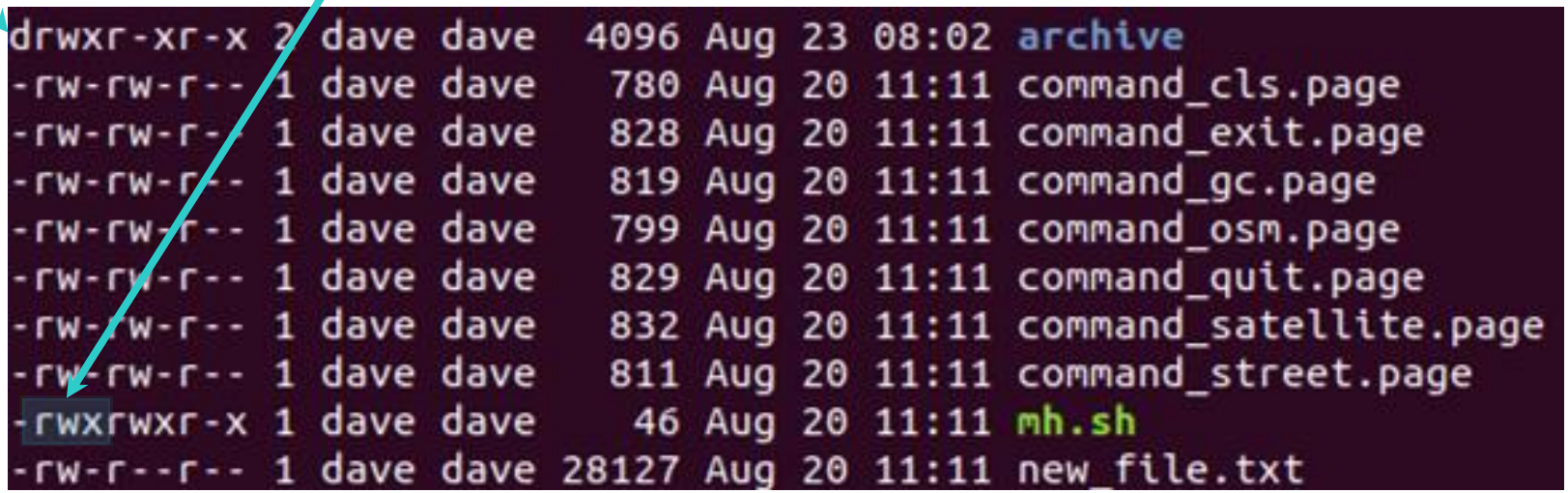
chmod **100** *file* - execute by user
chmod 010 *file* - execute by group
chmod 001 *file* - execute by other (World)

- So; 400 + 200 + 100 = 700
- Thus; 700 means **Read/Write/Execute** by **USER** only
- [chmod 777 = give full access to everyone]

Viewing Permissions

- The letters represent
  - *r*: Read permissions. (opened and viewed)
  - *w*: Write permissions. (edited, modified, and deleted)
  - *x*: Execute permissions. (program can be run)
  - -: No permission has been granted
  - Thus; **rwx** means full permissions have been granted by collective

Folder (directory)



System Management Commands (some)

- **df** — Displays disk free space on your system
- **free** — Displays RAM (used and free)
- **ip** — Displays network details, can also be used to configure network-related settings
- **ps** — Displays currently running processes
- **whoami** — Displays the current user name
- **mount**/**umount** — Attaches and detaches a separate filesystem (e.g. hard drives or USB)
- **kill**/**killall** — Use to end a process according to its process ID (often used in conjunction with the ps command) whereas you can use killall to end all processes whose names match your query.

System Management Commands (other)

- **Need to know of……..**
- Install new packages, upgrade packages, remove packages, etc.
- **apt** (advanced package tool) – although it isn't a command in itself, there are three commands that you must know to make full use of **apt**:
  - **add-apt-repository** (for locating third-party packages)
  - **apt-get** (for actually installing packages)
  - **apt-cache** (for searching your repositories)
  - [If your Linux version doesn't use APT, it may use YUM, RPM, or some other alternative]

Linux Scripts

- Similar to Microsoft Batch files (.bat) Linux shell scripts can be created and executed (.sh)
- Stages
  - **Create a file using** a the **vi** editor (or any other-editor). And Name the script file with **extension .sh**
  - Start the script with **#!/bin/sh**
    - ["#!" is an operator called shebang which directs the script to the interpreter location]
  - Write some code (script)
  - Save the script file as **filename.sh**
  - For **executing** the script type **bash filename.sh**
- Example script

```
#!/bin/sh
echo “What is your name?"
read name
echo "How do you do, $name“
```

Summary

- Another OS but FREE
- Selection of OS should be made and depend on user needs
- Linux uses a Familiar file and directory structure
- Command line commands permit navigation, file control and system management
- Linux uses an effective simple file permission system
- Shell scripts can be written that execute similar to that of (windows) batch files.

Workshop

- Complete Linux Workshop/Assignment material
- Submission date is the end of module.