Session 03:

Linux :

**Linux** was **inspired by Unix** and can be considered a **Unix-like operating system**

Unix was not free, it was for commercial purposes.

1991, the student called **Linus Trivolds** released the first version of Linux Kernal , it was just a kernal (Central component of OS, **kernel** acts as an intermediary between the user applications and the underlying hardware) , not complete OS. Linux was open source and made it available for everyone.

Later there are different Linux flavours formed from the Linux base kernal like Redhat, Centos, Ubuntu , Fedora etc.

Linux =Kernal

Linux OS = Kernal + Software

Command Shutdown the linux system :

Init 0

Restart

Init 6

Package Manager :

Ubuntu

The package manager in Ubuntu is called **APT** (Advanced Package Tool), and it's used to handle the installation, removal, and management of software packages on a system. APT makes it easy to install software from repositories and handle dependencies automatically.

Here are some basic examples of how to use APT for various tasks:

**1. Update package list:**

Before installing or upgrading packages, it's a good idea to update your package list to ensure you have the latest information about available software.

sudo apt update

**2. Upgrade installed packages:**

Once the package list is updated, you can upgrade all the installed packages on your system.

sudo apt upgrade

If you want to upgrade the system without being prompted for confirmation for each package, you can use:

sudo apt upgrade -y

**3. Install a package:**

To install a package, use the apt install command. For example, to install the text editor vim:

sudo apt install vim

You can install multiple packages at once by separating them with spaces:

sudo apt install vim git curl

**4. Remove a package:**

To remove a package, use the apt remove command. For example, to remove vim:

sudo apt remove vim

If you want to remove the package and its configuration files (completely), use the purge command:

sudo apt purge vim

**5. Search for a package:**

You can search for packages available in the repositories using the apt search command. For example, to search for vim:

apt search vim

**6. Show information about a package:**

To display detailed information about a package, use the apt show command. For example, to show information about vim:

apt show vim

**7. Clean up unused packages:**

To remove packages that were automatically installed as dependencies but are no longer needed, use:

sudo apt autoremove

Additionally, to clean the local cache of downloaded package files, use:

sudo apt clean

**8. Upgrade the entire system:**

To upgrade your Ubuntu system to the latest release (e.g., from Ubuntu 20.04 to Ubuntu 22.04), you can use the do-release-upgrade command:

sudo do-release-upgrade

RedHat

the equivalent package management example using **YUM** (for older versions of RHEL/CentOS) or **DNF** (for newer versions like RHEL 8 and CentOS 8) for a Red Hat-based system

sudo dnf install docker

sudo yum install docker

Almost all the commands are same in all linux OS except package manager commands

Onworks.net – web for linux GUI

Root Directory

/ (forward slash)

A screen shot of a computer

AI-generated content may be incorrect.

 **/**: Root directory; the top-level directory that contains the entire filesystem.

 **/bin**: Essential command binaries (e.g., ls, cp) needed for system operation. (which cp - to find the path from where ‘cp’ is executing

 **/sbin**: System binaries required for system administration and recovery. Eg: init (which init) , high privileged commands

 **/boot**: Files required for the boot process, including the Linux kernel and bootloader.

 **/dev**: Device files representing hardware (e.g., /dev/sda for storage).

 **/etc**: System-wide configuration files for applications and system settings.

 **/home**: User home directories where personal files and settings are stored (e.g., /home/user).

 **/root**: Home directory for the root user (superuser).

 **/lib**: Shared libraries and kernel modules required for system programs to run.

 **/media**: Mount points for removable media (e.g., USB drives, CDs).

 **/mnt**: Temporary mount point for mounting file systems manually.

 **/opt**: Optional application software, often third-party software installations.

 **/proc**: Virtual filesystem providing information about system processes and hardware.

 **/run**: Data stored during system runtime (e.g., process IDs, sockets).

 **/srv**: Data for services provided by the system (e.g., web server files).

 **/sys**: Virtual filesystem providing system and kernel information.

 **/tmp**: Temporary files used by programs and processes during execution.

 **/usr**: User-related programs and libraries; typically includes software and documentation.

 **/var**: Variable data files like logs, mail, and database files that change frequently.

Partitions can be created under root directory /mnt ,

This is not similar to windows directory structure , in windows there is no direct link b/w drives

Root and normal user login :

[root@Server01 ~]# - root user(admin user)

[user1@Server01 ~]$ - user login

~ : default directory of user (/root or /home)

[root@Server01 ~]# ls -l / or ll / - to find the existing file system (-l : long list)

If first character is ‘d’ , those are directories

If first character is ‘l’ , those are linked files (just like shortcuts)

If first character is ‘-‘ , those are just files

Root user native directory is /root (all secrets , high privilege permissions)

Normal users native directory is /home

cd <directory name> – change directory

cd .. – back to the parent directory of your current directory

cd /directory - move to specific directory

cd / - switch to root directory (eg: cd /etc , cd /etc/yum/vars/ )

cd : just cd will take us to user home directory

cd - : change the directory back to the previous directory you were in

sudo -i : Move to the root user

exit – go back to the normal user

compgen -c : all available commands

Commands are case sensitive in linux

man – help (man <command>)