

## ***Module :2- Linux server - Operate running systems***

### **1. View running processes with ps.**

**Ans:** - The ps command displays active processes running on your system.

- To see all processes for the current user: - ps aux
- To see all processes on the system: - ps -e
- To filter processes by name (e.g., SSH): - ps aux | grep ssh

### **2. Terminate processes with kill.**

**Ans:** - You can use kill to terminate a process by its PID (Process ID).

1. Find the PID using ps or top: - ps aux | grep process\_name
2. Kill the process using: - kill PID
3. If a process does not terminate, force it with: - kill -9 PID

### **3. Use top or htop to monitor system resources and processes.**

**Ans:** - Run top (default process monitor): - Top

Press q to exit.

② Run htop (better visual display, requires installation): -htop

- If htop is not installed, install it using: - sudo apt install htop

sudo yum install htop.

**4. Configure one of your lab COMPUTERS to boot to the CLI using systemd, and reboot to confirm that you were successful.**

**Ans:** - To configure your system to boot into CLI (text mode) instead of GUI:

**1. Set the default target to multi-user (CLI mode):**

sudo systemctl set-default multi-user.target

**2. Reboot the system to apply changes:**

sudo reboot

**3. Confirm that the system boots into CLI mode.**

If you ever want to switch back to GUI mode, use:

sudo systemctl set-default graphical.target

## ***Module :3- Linux server - Configure local storage Assignment***

### **1. Learn about different filesystem types (e.g., ext4, NTFS)**

**Ans:** - A filesystem in Linux is a way to organize, store, and retrieve data on storage devices. Linux supports multiple filesystems, each designed for different use cases.

#### **1. ext4 (Fourth Extended Filesystem)**

- Default filesystem in most Linux distributions (Ubuntu, Debian, CentOS).
- Features:
  - Journaling for faster recovery after crashes.
  - Supports files up to 16TB and volumes up to 1 exabyte (EB).
  - Backward compatible with ext3 and ext2.
  - Extents-based storage for better performance.
  - Delayed allocation to reduce fragmentation.
- Use case: General-purpose Linux systems.

#### **2. XFS (Extent File System)**

- High-performance journaling filesystem.
- Used by RHEL, CentOS, and SUSE Linux Enterprise.
- Features:
  - Handles large files and large storage (maximum 8 exabytes).

- High scalability and supports parallel I/O operations.
- Online resizing supported (only for growing, not shrinking).
- Use case: Enterprise environments, high-performance computing.

### **3. Btrfs (B-Tree File System)**

- Advanced, modern filesystem with features similar to ZFS.
- Features:
  - Copy-on-write (CoW) for data integrity.
  - Built-in RAID support (RAID 0, 1, 5, 6, 10).
  - Snapshots and subvolumes.
  - Online resizing (shrink and grow).
- Use case: Fileservers, data storage with redundancy.

### **4. ZFS (Zettabyte File System)**

- Developed by Sun Microsystems.
- Features:
  - Built-in RAID-Z support.
  - Data integrity with checksums.
  - Supports deduplication and compression.
  - Snapshots and cloning.
- Use case: Large-scale storage, enterprise storage solutions.

### **5. NTFS (New Technology File System)**

- Used primarily in Windows OS but supported in Linux via ntfs-3g driver.

- Features:
  - Supports large files (up to 16 exabytes).
  - Journaling for crash recovery.
  - File-level security with ACLs.
- Use case: External storage drives, dual-boot systems.

## 6. FAT32 (File Allocation Table)

- Legacy filesystem, widely used for USB drives and memory cards.
- Features:
  - Compatible with almost all operating systems.
  - File size limit of 4GB.
- Use case: USB drives, memory cards, cross-platform compatibility.
- 

## 7. exFAT (Extended File Allocation Table)

- Improved FAT filesystem for flash storage.
- Features:
  - No journaling.
  - Supports large files (up to 16EB).
- Use case: USB drives, SD cards with large files.

## 8. ReiserFS (Reiser File System)

- Used in older Linux distributions like SUSE.

- **Features:**
  - Journaling support.
  - Efficient small file storage.
- **Use case:** older systems, niche applications.

## 9. tmpfs (Temporary File System)

- Stores files in RAM instead of disk.
- Features:
  - Fast read/write speeds.
  - Data is lost after reboot.
- Use case: Temporary storage, /tmp directory.

## 2. Manage disk partitions and filesystems using tools like fdisk, mkfs, and mount.

**Ans:** - fdisk: Used to create, delete, and manage partitions.

- mkfs: Formats a partition with a specified filesystem (e.g., mkfs. ext4).
- mount: Attaches a filesystem to a directory.
- umount: Detaches a filesystem.

**3.** create a 2048MB partition and verify if the partition has been created.

**Ans:** - Open fdisk: - sudo fdisk /dev/sdX

- Create a new partition:
  - Press n → Enter (for new partition)
  - Choose primary or extended (p for primary)
  - Select partition number (default)
  - Specify start and end (+2048M for 2048MB)
  - Press w to save.
- Verify partition: - lsblk

sudo fdisk -l

**4.** Why LVM is required?

**Ans:** - LVM (Logical Volume Manager) allows flexible partition management:

- Resize volumes dynamically without unmounting.
- Combine multiple disks into a single volume.
- Take snapshots of a volume.
- Create thin-provisioned volumes.

**5. How can you find out how much memory Linux is using?**

**Ans:** - You can check memory usage with:

free -h

or

cat /proc/meminfo

or

vmstat -s

**6. What is a typical size for a swap partition under a Linux system?**

**Ans:** -If RAM is less than and equal to 2GB ----Swap =  $2 \times$  RAM.

- If RAM is 2GB - 8GB ----Swap = RAM.
- If RAM is greater than and equal to 8GB ----- Swap = 2GB or more (optional).

**7. What is the maximum file size on the ext4 file system?**

**Ans:** - 16TB (for a 64-bit system).

**8. What is the maximum file size on the xfs file system?**

**Ans:** - 8 exabytes (theoretically), but practical limits depend on system architecture.

## **Module: 4- Linux server - Manage user and Groups and working with file systems**

- 1. Manage users and groups with commands like useradd, userdel, groupadd, and passwd**

**Ans:** - Managing Users: -

- Adding a New User (useradd)

The useradd command creates a new user in Linux.

Syntax: - useradd [options] username

- Setting or Changing a User Password (passwd)

The passwd command is used to set or change a user's password.

Syntax: passwd username

- Deleting a User (userdel)

The userdel command removes a user account.

Syntax: userdel [options] username

**2. Managing Groups**

- Creating a Group (groupadd)

- Adding a User to a Group (`usermod -aG`) -The `usermod` command modifies user accounts. The `-aG` option adds a user to a secondary group.

```
usermod -aG groupname username
```

- Removing a User from a Group (`gpasswd -d`)

The `gpasswd -d` command removes a user from a group.

Syntax: `gpasswd -d username groupname`

- Deleting a Group (`groupdel`) -The `groupdel` command deletes a group.

Syntax: `groupdel groupname`

## 2. Explain different file system types in Linux?

**Ans:** -1. ext Family (Extended File System)

These are the most widely used file systems in Linux.

a. ext2 (Second Extended File System)

- Introduced in 1993.
- Lacks journaling (which means it is slower to recover after crashes).

- Used in USB flash drives and embedded systems.

b. ext3 (Third Extended File System)

- Introduced journaling, improving reliability.
- Backward compatible with ext2.
- Slower than modern file systems like ext4.

c. ext4 (Fourth Extended File System)

- Default file system for most Linux distributions.
- Supports larger file sizes (up to 16TB).
- More efficient journaling and performance improvements over ext3.

## 2. XFS (eXtents File System)

- High-performance journaling file system developed by SGI.
- Suitable for large-scale storage with high throughput.
- Used in enterprise environments, like RHEL.

## 3. Btrfs (B-Tree File System)

- Developed by Oracle.
- Features:
  - Copy-on-Write (CoW) mechanism.
  - Snapshots and built-in RAID support.
  - Self-healing capabilities.
- Suitable for advanced storage solutions.

## 4. ZFS (Zettabyte File System)

- Originally developed by Sun Microsystems.

- Features:
  - Data integrity checks and automatic repair.
  - High scalability (up to exabytes of data).
  - Integrated volume management.
- Used in NAS (Network Attached Storage) solutions.

## 5. ReiserFS

- Developed for fast file handling.
- Supports journaling.
- Less commonly used today due to lack of development.

## 6. JFS (Journaled File System)

- Developed by IBM.
- Optimized for performance and low CPU usage.
- Suitable for large file systems with minimal overhead.

## 7. FAT Family (FAT32, exFAT, and NTFS)

- FAT32: Used for USB drives, but has a 4GB file size limit.
- exFAT: A better alternative for external drives; supports large file sizes.
- NTFS: Developed by Microsoft but supported in Linux via ntfs-3g.

## 8. Swap File System

- Used for virtual memory (swap space).
- Helps when RAM is full by using disk space.

## 9. Network File Systems

- NFS (Network File System): Used for sharing files over a network.
- SMB/CIFS (Samba): Allows Linux to access Windows shared files.

## 10. Special Purpose File Systems

- tmpfs: Stores files in RAM for faster access.
- procfs: Used for accessing kernel-related information (/proc directory).
- sysfs: Exposes system device information (/sys directory).

### 3. Explain File Permission groups in Linux?

**Ans:** - In Linux, file permissions are crucial for security and access control. Permissions are assigned to three different groups:

#### 1. Owner (User)

- The user who owns the file.
- Typically, the creator of the file.
- Can modify permissions for others.

#### 2. Group

- A collection of users who share access rights.
- Any user within the group can have specific permissions.
- Useful for team collaboration.

### 3. Others (World)

- Any user on the system who is neither the owner nor in the group.
- Defines access for all other users.
- Viewing File Permissions

Use the ls -l command to check permissions: - ls -l file.txt

Output:

-rw-r--r-- 1 user group 1234 Mar 31 10:00 file.txt

- Explanation:
  - -rw-r--r--
    - - File type (- for a file, d for directory).
    - rw- Owner has read (r) and write (w) permissions.
    - r-- Group has read (r) permission.
    - r-- Others have read (r) permission.
- Changing Permissions

Use the chmod command: - chmod 764 file.txt

- 7 (Owner): Read (4) + Write (2) + Execute (1) = rwx
- 6 (Group): Read (4) + Write (2) = rw-
- 4 (Others): Read (4) = r--

Now, ls -l would show: -rwxrw-r-- 1 user group 1234 Mar 31 10:05 file.txt

#### **4. How do you switch from one desktop environment to another, such as switching from KDE to Gnome?**

**Ans:** - Switching from one desktop environment (DE) to another, such as KDE to GNOME, depends on whether you want to switch temporarily for a session or set a different DE as the default.

##### **1. Switching Desktop Environments Temporarily (For One Session)**

If you have both KDE and GNOME installed, you can switch between them from the login screen:

On Ubuntu (GDM Login Manager)

1. Log out of your current session.
2. At the login screen, click on your username.
3. Look for a gear icon  or a menu option (usually in the bottom-right corner).
4. Select GNOME (or "GNOME on Xorg" if available).
5. Enter your password and log in.

On KDE's SDDM Login Manager

1. Log out of KDE.
2. On the login screen, look for a "Session" button (usually a dropdown menu).
3. Select GNOME.
4. Log in.

## 2. Setting GNOME as the Default Desktop Environment

- If you want GNOME to be the default DE, follow these steps:

For Display Manager (GDM or SDDM)

### 1. Ensure GNOME is installed

Run the following command to install GNOME if it isn't installed:

```
sudo apt install gnome-session or on Fedora:  
- sudo dnf install gnome-session
```

### 2. Set GNOME as Default

- If you're using GDM, it automatically remembers your last selection.
- If you're using SDDM (from KDE), you may need to change the configuration:

```
sudo nano /etc/sddm.conf
```

Look for: -Session=plasma.desktop

Change it to: - Session=gnome.desktop

Save and exit (Ctrl + X, then Y to confirm).

### 3. Restart the system: - sudo reboot

Next time you log in, GNOME should be the default.

### 3. Removing KDE (Optional)

If you want to completely remove KDE, use:

```
-sudo apt remove --purge kde-plasma-desktop  
sudo apt autoremove
```

For Fedora:

```
- sudo dnf group remove "KDE Plasma Workspaces"
```

## 5. . What are the kinds of permissions under Linux?

Ans: - In Linux, file and directory permissions determine who can read, write, or execute a file. There are three types of permissions and three categories of users that these permissions apply to.

### Types of Permissions

1. Read (r) - Allows viewing the contents of a file or listing a directory's contents.
2. Write (w) - Allows modifying the contents of a file or creating/deleting files within a directory.
3. Execute (x) - Allows executing a file (if it's a script or program) or accessing a directory.

### Categories of Users

1. Owner (User - u) - The user who owns the file.

2. Group (g) - Other users who belong to the file's assigned group.
3. Others (o) - All other users on the system.

## Permission Representation

Permissions can be represented in:

### 1. Symbolic Mode (rwx format)

Example: `-rwxr-xr--`

- The first character (`-`) represents the file type (`d` for directory, `-` for regular file).
- The next three characters (`rwx`) are for the owner.
- The next three (`r-x`) are for the group.
- The last three (`r--`) are for others.

### 2. Numeric Mode (Octal format) Each permission has a numeric value:

- Read (`r`) = 4
- Write (`w`) = 2
- Execute (`x`) = 1
- No permission = 0

Example:

- 777 - Full permissions for owner, group, and others (`rwxrwxrwx`).
- 644 - Read/write for owner, read-only for group and others (`rw-r--r--`).

## Special Permissions

## 1. SetUID (Set User ID) (s)

- Represented as s in the owner's execute field (rws).
- When applied to a file, it allows the file to run as the owner's permissions.
- Example: chmod 4755 file

## 2. SetGID (Set Group ID) (s)

- Represented as s in the group's execute field (rws).
- When applied to a directory, all new files inherit the group of the directory.
- Example: chmod 2755 directory

## 3. Sticky Bit (t)

- Represented as t in the others execute field (rwxrwxrwt).
- Applied to directories to prevent users from deleting other users' files.
- Example: chmod 1777 /tmp

## 6. . What are the different modes when using vi editor?

Ans: - The vi editor has three primary modes:

### 1. Normal Mode (Command Mode)

- This is the default mode when you open the vi editor.
- Used for navigating the file, deleting text, copying, pasting, searching, and other operations.
- You cannot insert text in this mode.
- Press `Esc` to return to Normal mode from any other mode.

Common commands in Normal mode:

- h - Move left
- l - Move right
- j - Move down
- k - Move up
- x - Delete a character
- dd - Delete a whole line
- yy - Copy a line (yank)
- p - Paste
- /Text - Search for "text"

## 2. Insert Mode

- Used for inserting and editing text.
- Enter this mode by pressing:
  - i - Insert before the cursor
  - I - Insert at the beginning of the line
  - a - Append after the cursor
  - A - Append at the end of the line
  - o - Open a new line below the cursor
  - O - Open a new line above the cursor
- Press `Esc` to return to Normal mode.

## 3. Command-Line Mode (Ex Mode)

- Used for saving, quitting, and performing advanced operations.
- Enter this mode by pressing: in Normal mode.
- Common commands:

- : w - Save the file
- : q - Quit
- : wq or ZZ - Save and quit
- : q! - Quit without saving
- : set nu - Show line numbers
- : set nonu - Hide line numbers

#### 4. Visual Mode (Selection Mode)

- Used to select text for copying, deleting, or replacing.
- Enter this mode by pressing:
  - v - Start selecting character by character
  - V - Select entire lines
  - Ctrl+v - Select a block (Column mode)
- After selecting, you can perform actions like:
  - y - Copy
  - d - Delete
  - p - Paste

