

[1] User Management in Linux:

Managing users is a critical task for maintaining system security and ensuring that only authorized users have access to the system. Here are some essential commands for user management:

- **Create a New User:**

To create a new user with a home directory:

```
sudo adduser username
```

- **Delete a User:**

To remove a user account:

```
sudo deluser username
```

To delete the user along with their home directory:

```
sudo deluser --remove-home username
```

- **Change a User's Password:**

To set or change the password for an existing user:

```
sudo passwd username
```

- You will be prompted to enter and confirm the new password.

- **Add a User to a Group:**

To add an existing user to a group:

```
sudo usermod -aG groupname username
```

- **List All Users:**

To display a list of all user accounts on the system:

```
cut -d: -f1 /etc/passwd
```

- **Check Groups a User Belongs To:**

To see the groups a user is part of:

```
groups username
```

- **Lock and Unlock a User Account:**

To lock a user account, preventing them from logging in:

```
sudo usermod -L username
```

To unlock the user account:

```
sudo usermod -U username
```

These commands are vital for managing user access and maintaining system security in a multi-user environment..

[2]Adding a New Storage Device (partition and format) Mount Storage Devices Mount Partitions Automatically During Boot (fstab) Monitoring Disk Space:

1. Identify the New Disk:

List all disks and partitions:

```
lsblk
```

- Identify the new disk (e.g., /dev/sdX).

2. Partition the Disk:

- Use fdisk for MBR partitions or parted for GPT partitions.

Example with fdisk:

```
sudo fdisk /dev/sdX
```

- Inside fdisk, you can use commands like:
 - n to create a new partition
 - p to print the partition table
 - w to write changes and exit

3. Format the Partition:

Format the partition with a filesystem, e.g., ext4:

```
sudo mkfs.ext4 /dev/sdX1
```

Mount Storage Devices

1. Create a Mount Point:

Make a directory where the partition will be mounted:

```
sudo mkdir /mnt/mydisk
```

2. **Mount the Partition:**

Mount the partition to the directory:

```
sudo mount /dev/sdX1 /mnt/mydisk
```

3. **Verify the Mount:**

Check the mounted filesystems:

```
df -h
```

Mount Partitions Automatically During Boot (fstab)

1. **Edit the /etc/fstab File:**

Open the file in a text editor:

```
sudo nano /etc/fstab
```

2. **Add an Entry for the Partition:**

Add a line for the new partition in the following format:

```
/dev/sdX1 /mnt/mydisk ext4 defaults 0 2
```

3. **Test the Configuration:**

To test the changes without rebooting:

```
sudo mount -a
```

Monitoring Disk Space

1. **Check Disk Space Usage:**

To display disk space usage for all mounted filesystems:

```
df -h
```

- The -h flag shows sizes in a human-readable format (e.g., GB, MB).

2. **Check Disk Usage of Specific Directories:**

To see the disk usage of a specific directory:

```
du -sh /path/to/directory
```

- The -s flag summarizes the total size, and -h makes it human-readable.

3. **Monitor Disk Space Over Time:**

- For continuous monitoring, tools like df in a cron job or system monitoring tools like htop and glances can be useful.

These steps will help you manage storage devices effectively on a Linux system, from partitioning and formatting to mounting and monitoring.

[3]Resource management - Configure systemd units Get status of systemd units Start and stop services Enable / disable systemd services for runtime, etc:

Resource Management with systemd

Systemd is a system and service manager for Linux operating systems, providing a standardized way to manage services and resources. Here's a guide on configuring and managing systemd units:

1. Configure systemd Units

Systemd units are configuration files that define services, mount points, devices, and other system resources. They are typically located in /etc/systemd/system/ for custom units and /lib/systemd/system/ for system-provided units.

1. Create or Edit a Systemd Unit File:

To create or modify a unit file, you might use a text editor like nano or vim:

```
sudo nano /etc/systemd/system/myservice.service
```

2. Define Unit File Sections:

A basic service unit file might look like this:

```
ini
```

```
[Unit]
```

```
Description=My Custom Service
```

```
After=network.target
```

```
[Service]
```

```
ExecStart=/usr/bin/mycommand
```

```
Restart=always
```

[Install]

WantedBy=multi-user.target

- **Sections:**

- [Unit]: Metadata and dependencies.
- [Service]: Service-specific options (e.g., what command to run).
- [Install]: Installation information (e.g., when to start the service).

3. **Reload Systemd Configuration:**

After creating or modifying a unit file, reload systemd to apply changes:

```
sudo systemctl daemon-reload
```

2. **Get Status of Systemd Units**

1. **Check the Status of a Service:**

To see the status of a specific service:

```
sudo systemctl status myservice
```

- This command shows whether the service is running, its recent logs, and other details.

2. **List All Active Services:**

To view all currently active services:

```
sudo systemctl list-units --type=service
```

3. **List All Systemd Units:**

To view all systemd units (not just services):

```
sudo systemctl list-units
```

3. **Start and Stop Services**

1. **Start a Service:**

To manually start a service:

```
sudo systemctl start myservice
```

2. **Stop a Service:**

To stop a currently running service:

```
sudo systemctl stop myservice
```

3. **Restart a Service:**

To restart a service, which is useful if you have made changes:

```
sudo systemctl restart myservice
```

4. **Reload a Service:**

To reload the service configuration without stopping the service:

```
sudo systemctl reload myservice
```

4. **Enable / Disable Systemd Services**

1. **Enable a Service:**

To enable a service to start automatically on boot:

```
sudo systemctl enable myservice
```

2. **Disable a Service:**

To prevent a service from starting at boot:

```
sudo systemctl disable myservice
```

3. **Check if a Service is Enabled:**

To check if a service is set to start on boot:

```
sudo systemctl is-enabled myservice
```

4. **Mask a Service:**

To prevent a service from being started manually or automatically:

```
sudo systemctl mask myservice
```

5. **Unmask a Service:**

To undo the masking of a service:

```
sudo systemctl unmask myservice
```

These commands and configurations allow you to manage services and resources effectively using systemd, making it easier to control and automate system behavior.