**Module:7- Linux server -deployment of network services**

**57. Difference between LILO and GRUB**

LILO (Linux Loader) and GRUB (GRand Unified Bootloader) are both boot loaders used in Linux systems, but they have several key differences:

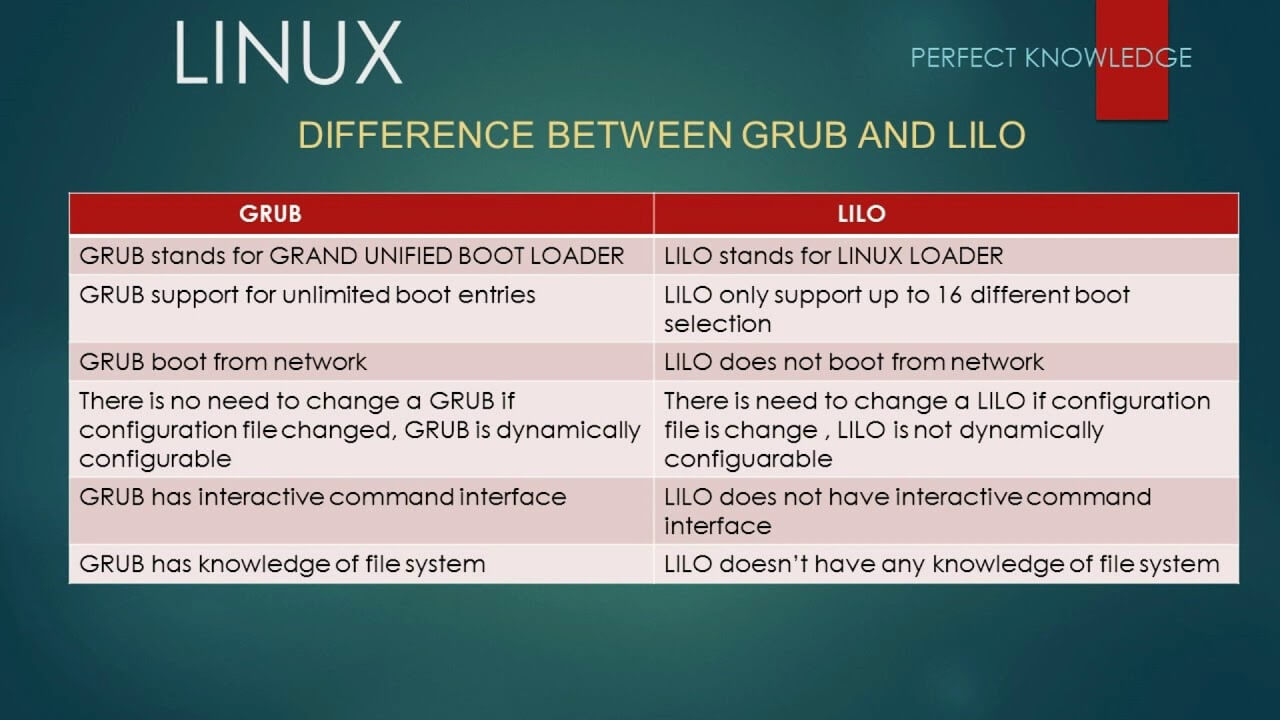
Multiboot Support: GRUB supports multiboot, allowing you to boot multiple operating systems on a single machine. LILO, on the other hand, is limited to booting Linux and a few other operating systems12.

Graphical User Interface: GRUB provides a graphical user interface (GUI) menu, making it easier to select the desired operating system or kernel version during boot. LILO lacks a GUI and operates through a command-line interface12.

Filesystem Support: GRUB supports a wide range of filesystems, including ext4, Btrfs, XFS, and more. LILO requires a separate configuration file for each kernel image and does not support as many filesystems12.

Configuration Changes: GRUB allows dynamic changes to the configuration during boot, meaning you can modify boot parameters on the fly. LILO requires you to rewrite the boot loader to the Master Boot Record (MBR) after any configuration changes12.

Network Booting: GRUB supports booting from a network, which is useful for diskless systems. LILO does not have this capability12.



**58. How to Recover Linux Password**

If you forget your Linux password:

Method 1: Using GRUB

Reboot the System: Restart your computer and hold the Shift key to access the GRUB menu.

Edit Boot Options: Highlight the default boot option and press ‘e’ to edit.

Modify Boot Parameters: Find the line starting with linux and append init=/bin/bash at the end.

Boot into Single-User Mode: Press Ctrl + X or F10 to boot.

Remount Filesystem: Once in the shell, remount the filesystem in read/write mode using mount -o remount,rw /.

Change Password: Use the passwd command to change the root password: passwd root.

Reboot: Remount the filesystem in read-only mode using mount -o remount,ro / and then reboot using reboot34.

Method 2: Using a Live CD/USB

Boot from Live CD/USB: Insert a Linux live CD/USB and boot from it.

Open Terminal: Open a terminal window.

Mount Root Filesystem: Identify your root partition (e.g., /dev/sda1) and mount it: mount /dev/sda1 /mnt.

Change Root: Change root to the mounted filesystem: chroot /mnt.

Change Password: Use the passwd command to change the root password: passwd root.

Reboot: Exit the chroot environment and reboot: exit and reboot3

**59. Command to Format Partition in Linux OS**

To format a partition in Linux, you can use the mkfs (make filesystem) command.

Step-by-Step:

Identify the Partition: Use lsblk or fdisk -l to list all partitions and identify the one you want to format.

Format the Partition: Use the mkfs command with the appropriate filesystem type.

Examples:

Format with ext4:

sudo mkfs.ext4 /dev/sdXn

Replace /dev/sdXn with your partition identifier (e.g., /dev/sda1).

Format with FAT32:

sudo mkfs.vfat /dev/sdXn

Format with NTFS:

sudo mkfs.ntfs /dev/sdXn

Format with XFS:

sudo mkfs.xfs /dev/sdXn

After formatting, you can mount the partition using the mount command and create a mount point if necessary:

sudo mount /dev/sdXn /mnt/your\_mount\_point

**60. How to Enable “Quota” in Linux**

Disk quotas are used to limit the amount of disk space and the number of files a user or group can use.:

Install Quota Package:

sudo apt-get install quota

Edit /etc/fstab: Add usrquota and/or grpquota options to the desired partition. For example:

/dev/sda1 /home ext4 defaults,usrquota,grpquota 0 2

Remount the Filesystem:

sudo mount -o remount /home

Create Quota Files:

sudo quotacheck -cum /home

sudo quotaon -v /home

Set Quotas for Users:

sudo edquota -u username

This will open a file where you can set the soft and hard limits for blocks and inodes.

Verify Quotas:

sudo repquota /home

**61. How to Mount Partition in Linux**

Mounting a partition in Linux involves attaching a filesystem to a directory. Here’s how you can do it:

Identify the Partition:

sudo fdisk -l

Create a Mount Point:

sudo mkdir /mnt/my\_partition

Mount the Partition:

sudo mount /dev/sda1 /mnt/my\_partition

Verify the Mount:

df -h

Permanent Mount (Optional): Edit /etc/fstab to make the mount permanent:

/dev/sda1 /mnt/my\_partition ext4 defaults 0 2

**62. What is the Use of “mdadm” Command?**

The mdadm command is used for managing and monitoring RAID (Redundant Array of Independent Disks) arrays in Linux. It allows you to create, assemble, monitor, and manage RAID arrays, ensuring data redundancy and improved performance. Here are some key uses:

Creating RAID Arrays: Set up different RAID levels (e.g., RAID 0, RAID 1, RAID 5) to combine multiple disks into a single array.

Assembling RAID Arrays: Reassemble existing RAID arrays after a reboot or system crash.

Monitoring RAID Arrays: Check the status and health of RAID arrays to ensure they are functioning correctly.

Adding/Removing Devices: Add new disks to an existing array or remove faulty ones.

Checking RAID Status: Use /proc/mdstat to view the current status of all RAID arrays.

Managing Configuration: Update the mdadm configuration file to ensure arrays are recognized at boot.

Install mdadm:

sudo apt-get install mdadm

Create a RAID Array:

sudo mdadm --create --verbose /dev/md0 --level=1 --raid-devices=2 /dev/sda1 /dev/sdb1

Monitor the RAID Array:

sudo mdadm --detail /dev/md0

Add Configuration to mdadm.conf:

sudo mdadm --detail --scan | sudo tee -a /etc/mdadm/mdadm.conf

Update Initramfs:

sudo update-initramfs -u

Check RAID Status:

cat /proc/mdstat

**63. How to Configure a Secure Apache Web Server in Linux**

Configuring a secure Apache web server involves several steps to ensure both functionality and security.

Update and Upgrade Your System:

sudo apt update

sudo apt upgrade

Install Apache:

sudo apt install apache2

Configure the Firewall:

sudo ufw allow 'Apache Full'

sudo ufw enable

Create a Virtual Host File:

sudo nano /etc/apache2/sites-available/your\_domain.conf

Add the following configuration:

<VirtualHost \*:80>

ServerAdmin admin@your\_domain

ServerName your\_domain

ServerAlias www.your\_domain

DocumentRoot /var/www/your\_domain

ErrorLog ${APACHE\_LOG\_DIR}/error.log

CustomLog ${APACHE\_LOG\_DIR}/access.log combined

</VirtualHost>

Enable the Virtual Host:

sudo a2ensite your\_domain.conf

sudo systemctl reload apache2

Install Certbot and Obtain SSL Certificate:

sudo apt install certbot python3-certbot-apache

sudo certbot --apache -d your\_domain -d www.your\_domain

Verify SSL Configuration:

sudo apache2ctl configtest

sudo systemctl reload apache2

Set Up Automatic Renewal: Certbot automatically renews certificates, but you can test it:

sudo certbot renew --dry-run

**64. How to Run Windows Software on Linux Operating System**

Running Windows software on Linux can be achieved using several methods.

Using Wine:

Install Wine:

sudo apt update

sudo apt install wine

Run Windows Software:

wine your\_windows\_program.exe

Using PlayOnLinux:

Install PlayOnLinux:

sudo apt update

sudo apt install playonlinux

Run PlayOnLinux and follow the GUI to install and manage Windows applications.

Using a Virtual Machine:

Install VirtualBox:

sudo apt update

sudo apt install virtualbox

Create a New Virtual Machine and install Windows on it. You can then run Windows software within this virtual machine.

Using CrossOver:

Install CrossOver (a commercial product based on Wine):

sudo apt update

sudo apt install crossover

Run Windows Software using CrossOver’s GUI.

**65. What is the Difference Between Windows and Linux**

comparison of Windows and Linux:

Source Code:

Linux: Open-source, free to modify and distribute.

Windows: Proprietary, source code is not available to the public.

Cost:

Linux: Generally free.

Windows: Requires a paid license.

Kernel Type:

Linux: Monolithic kernel.

Windows: Hybrid kernel.

User Privileges:

Linux: Root user has all administrative privileges.

Windows: Administrator user has all administrative privileges.

Case Sensitivity:

Linux: Case-sensitive file system.

Windows: Case-insensitive file system.

Security:

Linux: Generally more secure due to its permission and user management system.

Windows: More vulnerable to malware and viruses.

File System:

Linux: Uses a hierarchical file system where everything is a file.

Windows: Uses drive letters (C:, D:, etc.) and a directory structure.

Customization:

Linux: Highly customizable, with various desktop environments and configurations.

Windows: Limited customization options compared to Linux.

Software Installation:

Linux: Uses package managers (e.g., APT, YUM) and repositories.

Windows: Uses executable files (.exe) and installers.

Performance:

Linux: Generally more efficient and stable, especially for servers.

Windows: Can slow down over time, especially with prolonged use.

User Interface:

Linux: Multiple desktop environments (e.g., GNOME, KDE).

Windows: Standardized interface with each version (e.g., Windows 10, Windows 11).

**66. What is the Advantage of Open Source?**

Open source software offers several advantages:

Cost-Effective:

Open source software is usually free, reducing the cost of software acquisition and licensing.

Flexibility and Customization:

Users can modify the source code to fit their specific needs, allowing for high customization.

Community Support:

A large community of developers contributes to and supports open source projects, leading to rapid bug fixes and feature enhancements.

Transparency:

The source code is open for inspection, which builds trust and allows users to verify the software’s security and functionality.

No Vendor Lock-in:

Users are not dependent on a single vendor for updates or support, providing more freedom and flexibility.

Security:

With many eyes on the code, security vulnerabilities are often identified and resolved quickly1.

**67. Install and Configure Web Servers like Apache**

Install and configure an Apache web server on a Linux system:

Update Your System:

sudo apt update

sudo apt upgrade

Install Apache:

sudo apt install apache2

Configure the Firewall:

sudo ufw allow 'Apache Full'

sudo ufw enable

Verify Apache Installation:

Open a web browser and navigate to http://localhost. You should see the Apache default page.

Create a Virtual Host File:

sudo nano /etc/apache2/sites-available/your\_domain.conf

Add the following configuration:

<VirtualHost \*:80>

ServerAdmin admin@your\_domain

ServerName your\_domain

ServerAlias www.your\_domain

DocumentRoot /var/www/your\_domain

ErrorLog ${APACHE\_LOG\_DIR}/error.log

CustomLog ${APACHE\_LOG\_DIR}/access.log combined

</VirtualHost>

Enable the Virtual Host:

sudo a2ensite your\_domain.conf

sudo systemctl reload apache2

Set Up SSL (Optional):

Install Certbot:

sudo apt install certbot python3-certbot-apache

Obtain an SSL certificate:

sudo certbot --apache -d your\_domain -d www.your\_domain

Verify SSL Configuration:

sudo apache2ctl configtest

sudo systemctl reload apache2

**68. Host a Simple Website and Configure Virtual Hosts**

To host a simple website and configure virtual hosts:

Create the Directory Structure:

sudo mkdir -p /var/www/your\_domain/public\_html

sudo chown -R $USER:$USER /var/www/your\_domain/public\_html

sudo chmod -R 755 /var/www

Create a Sample HTML File:

nano /var/www/your\_domain/public\_html/index.html

Add the following content:

HTML

<html>

<head>

<title>Welcome to Your Domain!</title>

</head>

<body>

<h1>Success! Your Apache Virtual Host is working!</h1>

</body>

</html>

Create a Virtual Host Configuration File:

sudo nano /etc/apache2/sites-available/your\_domain.conf

Add the following configuration:

<VirtualHost \*:80>

ServerAdmin admin@your\_domain

ServerName your\_domain

ServerAlias www.your\_domain

DocumentRoot /var/www/your\_domain/public\_html

ErrorLog ${APACHE\_LOG\_DIR}/error.log

CustomLog ${APACHE\_LOG\_DIR}/access.log combined

</VirtualHost>

Enable the Virtual Host:

sudo a2ensite your\_domain.conf

sudo systemctl reload apache2

Test the Configuration:

Open a web browser and navigate to http://your\_domain. You should see the sample HTML page.

**69.Install and Manage MySQL/MariaDB**

1. Update Your System

First, ensure your system is up to date:

sudo apt update

sudo apt upgrade

2. Install MySQL/MariaDB

You can choose to install either MySQL or MariaDB. Here’s how to install MariaDB:

Install MariaDB:

sudo apt install mariadb-server

Install MySQL:

sudo apt install mysql-server

3. Secure the Installation

Run the security script to improve the security of your database installation:

sudo mysql\_secure\_installation

Follow the prompts to set the root password, remove anonymous users, disallow root login remotely, remove test databases, and reload privilege tables.

4. Start and Enable the Database Service

Ensure the database service starts on boot:

sudo systemctl start mariadb

sudo systemctl enable mariadb

For MySQL, replace mariadb with mysql.

5. Log in to MySQL/MariaDB

Log in to the database server as the root user:

sudo mysql -u root -p

Enter the root password you set during the secure installation step.

6. Create a Database

Create a new database:

AI-generated code. (😒😒😒😒😒)

SQL

CREATE DATABASE my\_database;

7. Create a User and Grant Privileges

Create a new user and grant them privileges on the database:

SQL

CREATE USER 'my\_user'@'localhost' IDENTIFIED BY 'password';

GRANT ALL PRIVILEGES ON my\_database.\* TO 'my\_user'@'localhost';

FLUSH PRIVILEGES;

8. Verify the Database and User

Check the databases and user privileges:

SQL

SHOW DATABASES;

SHOW GRANTS FOR 'my\_user'@'localhost';

AI-generated code.

9. Manage Databases

You can now manage your databases by creating tables, inserting data, and querying data.

Create a Table:

SQL

USE my\_database;

CREATE TABLE my\_table (

id INT AUTO\_INCREMENT PRIMARY KEY,

name VARCHAR(100),

age INT

);

Insert Data:

SQL

INSERT INTO my\_table (name, age) VALUES ('John Doe', 30);

Query Data:

SQL

SELECT \* FROM my\_table;

AI-generated code. (😢😢)

Additional Management Tools

Using phpMyAdmin

For a graphical interface, you can use phpMyAdmin:

Install phpMyAdmin:

sudo apt install phpmyadmin

Configure Apache to Work with phpMyAdmin:

sudo ln -s /usr/share/phpmyadmin /var/www/html/phpmyadmin

Access phpMyAdmin: Open a web browser and navigate to http://localhost/phpmyadmin. Log in with your database credentials.

Using MySQL Workbench

Another powerful tool is MySQL Workbench:

Install MySQL Workbench:

sudo apt install mysql-workbench

Launch MySQL Workbench: Open MySQL Workbench from your applications menu and connect to your database server.