Understanding Operating Systems

An operating system (OS) is software that manages computer hardware and software resources. It provides a platform for running programs and offers common services for them.

Components of a System

CPU (Central Processing Unit): The brain of the computer responsible for executing instructions and processing data.
RAM (Random Access Memory): Main memory used to store data and instructions currently in use by the CPU.
Hard Disk: A non-volatile storage device for permanent data storage.

Data Storage

Data can be stored on the hard disk or in RAM. Hard disk storage is permanent, while RAM is temporary and volatile.

Interacting with the OS

To interact with the OS, users run or execute program files or commands. A program file contains instructions for a program, and a command is a text instruction given to the OS.

Three Ways to Interact:

GUI (Graphical User Interface): Visual interface for user interaction.
CLI (Command Line Interface): Text-based interface where users input commands.

WebUI (Web User Interface): Interface accessible through a web browser.

Linux Operating System

Linux is a Unix-like operating system known for stability, security, and flexibility. It is widely used in data centers, servers, and some desktops/laptops.

User Types in Linux

- Root Users: Have unlimited power.
- Non-Root Users: Have limited power; can't install software or add users.

Using `sudo`

To perform actions requiring root access, users use `sudo` before the command.

Interaction Modes

- GUI: Graphical interface for user interaction.
- CLI: Text-based interface for command input.
- WebUI: Web-based interface accessible through a browser.

Linux Commands

Basic Commands

ping

- Purpose: Check network connectivity.
- Example: `ping google.com`

free

- Purpose: Display system memory usage.
- Example: `free -m`

which

- Purpose: Locate the executable of a command.
- Example: `which ls`

gedit

- Purpose: Open a text editor.
- Example: `gedit filename`

whoami

- Purpose: Display the current user.
- Example: `whoami`

useradd

- Purpose: Add a new user.
- Example: `sudo useradd newuser`

passwd

- Purpose: Change user password.
- Example: `sudo passwd username`

chvt4

- Purpose: Switch to virtual terminal 4.
- Example: `chvt4`

who

- Purpose: Display logged-in users.
- Example: `who`

tty

- Purpose: Print the file name of the terminal connected to standard input.
- Example: `tty`

history

- Purpose: Display command history.
- Example: `history`

man date

- Purpose: Display manual page for the `date` command.
- Example: `man date`

date +%h

- Purpose: Display the abbreviated month.
- Example: `date +%h`

date +%r

- Purpose: Display the time in 12-hour clock format.
- Example: `date +%r`

gnome-screenshot

- Purpose: Take a screenshot.
- Example: `gnome-screenshot`

gnome-screenshot –file=/root/ –delay=30 &

- Purpose: Take a delayed screenshot and save it to a specified location.
- Example: `gnome-screenshot --file=/root/ --delay=30 &`

pwd

- Purpose: Print the current working directory.
- Example: `pwd`

ls

- Purpose: List files and directories.
- Example: `ls`

jobs

- Purpose: Display status of jobs.
- Example: `jobs`

man rpm

- Purpose: Display manual page for the `rpm` command.
- Example: `man rpm`

rpm -q firefox

- Purpose: Query installed version of Firefox.
- Example: `rpm -q firefox`

rpm -q -i firefox

- Purpose: Query and display information about the installed Firefox package.
- Example: `rpm -q -i firefox`

rpm -q -a

- Purpose: Query all installed packages.
- Example: `rpm -q -a`

which date

- Purpose: Display the path to the `date` command.
- Example: `which date`

rpm -q -l firefox

- Purpose: List files installed by the Firefox package.
- Example: `rpm -q -l firefox`

All the devices in Linux OS are managed by /dev folder

- Purpose: View device files in the `/dev` directory.
- Example: `ls /dev`

df -h

- Purpose: Display disk space usage.

- Example: `df -h`

ls firefox-

- Purpose: List files with names starting with "firefox-".
- Example: `ls firefox-*`

rpm -e firefox

- Purpose: Remove the Firefox package.
- Example: `rpm -e firefox`

rpm -i firefox-91.8.0-1.el9_0.x86_64.rpm

- Purpose: Install a specific version of Firefox.
- Example: `rpm -i firefox-91.8.0-1.el9_0.x86_64.rpm`

rpm -ivh firefox-91.8.0-1.el9_0.x86_64.rpm

- Purpose: Install a specific version of Firefox with verbose output.
- Example: `rpm -ivh firefox-91.8.0-1.el9_0.x86_64.rpm`

ping localhost

- Purpose: Check network connectivity to localhost.
- Example: `ping localhost`

ifconfig

- Purpose: Display network interface configuration.
- Example: `ifconfig`

exit

- Purpose: Exit the current shell or terminal.
- Example: `exit`

ifconfig enp0s3 192.168.12345

- Purpose: Set a specific IP address for a network interface.
- Example: `ifconfig enp0s3 192.168.12345`

nslookup www.google.com

- Purpose: Query DNS for information about a domain.
- Example: `nslookup www.google.com`

netstat -tnlp

- Purpose: Display active network connections and listening ports.
- Example: `netstat -tnlp`

getenforce

- Purpose: Display the current SELinux enforcement mode.
- Example: `getenforce`

setenforce 0

- Purpose: Set SELinux enforcement mode to permissive.
- Example: `setenforce 0`

Virtualization Concept

- Oracle VirtualBox: A product for implementing virtualization.

Installation Methods

Directly on Hardware: Bare-metal installation.
Cloud Computing: Install on a virtual machine in the cloud.
Containerization: Install inside a container for isolation.
Virtualization: Install on a base OS (Windows, macOS).

Yum Package Manager

Yum is a package management utility for Linux, used for installing, updating, and managing software packages.

Features

Package Management
Dependency Resolution
Repository Support
Updates and Upgrades
User-Friendly Interface

Yum Configuration Steps

1. Mount the Drive

- Before configuring Yum, ensure that the installation media or repository is mounted.

2. Check Disk Space

```
df -h
```

- Verify available disk space, and note the path where the repository is mounted.

3. Copy the Path

- Copy the path of the mounted drive. This will be used in the Yum configuration.

4. Create Yum Repository Configuration

- Navigate to the `/etc/yum.repos.d/` directory.

```
cd /etc/yum.repos.d/
```

Create a new Yum repository configuration file. Replace "example.repo" with a relevant name.

```
gedit example.repo
```

Inside the file, add repository configurations. For instance:

```
[dvd1]

baseurl=file://<copied path>/AppStream

gpgcheck=0



[dvd2]

baseurl=file://<copied path>/BaseOS
```

```
gpgcheck=0
```

Save and exit the editor.

5. Verify Yum Repository

Check the configured Yum repositories:

```
yum repolist
```

Yum Commands

Install Software Package

```
yum install httpd
```

Remove Software Package

```
yum remove httpd
```

Additional Tips

- Yum automatically resolves dependencies during installation.
- To update the package list, use:

```
yum update
```

Yum keeps a cache of packages. To clean the cache, use:

```
yum clean all
```

Apache Web Server Configuration

Installation

Install the Apache web server using `yum install httpd`.
Configure the server by editing files in `/var/www/html/`.
Start the server using `systemctl start httpd`.
Enable automatic start on boot using `systemctl enable httpd`.
Verify the status with `systemctl status httpd`.

Firewall Configuration

- Check firewall status with `systemctl status firewalld`.
- Stop the firewall with `systemctl stop firewalld`.

Apache HTTP Server Configuration (httpd.conf)

The `httpd.conf` file is the main configuration file for the Apache HTTP Server. It contains settings that govern the behavior of the server, including directives related to ports, document roots, and other important concepts.

Location of httpd.conf

The `httpd.conf` file is typically located in the Apache configuration directory. On many Linux systems, the default location is:

```
/etc/httpd/conf/httpd.conf
```

Key Concepts in httpd.conf

1. Port Configuration

By default, Apache listens on port 80 for HTTP traffic. To configure a different port, locate the following directive in the `httpd.conf` file:

```
Listen 80
```

Replace `80` with the desired port number. For example, to use port 8080:

```
Listen 8080
```

Replace "/var/www/html" with the path to your desired document root.

2. Document Root

The DocumentRoot directive specifies the directory that contains the website's HTML files. It is the base directory against which requests for documents are mapped. Find and modify the DocumentRoot directive:

```
DocumentRoot "/var/www/html"
```

Replace "/var/www/html" with the path to your desired document root.

3. Virtual Hosts

Virtual Hosts allow multiple websites to be hosted on the same server. The `<VirtualHost>` directive in `httpd.conf` is used to configure each virtual host. Example:

```
<VirtualHost *:80>

    DocumentRoot "/var/www/site1"

    ServerName www.example.com

</VirtualHost>



<VirtualHost *:80>

    DocumentRoot "/var/www/site2"

    ServerName www.another-example.com

</VirtualHost>
```

Here, requests for `www.example.com` will be served from `/var/www/site1`, and requests for `www.another-example.com` will be served from `/var/www/site2`.

4. Directory Options

The `<Directory>` directive is used to apply configuration to a specific directory. Example:

```
<Directory "/var/www/html">

    Options Indexes FollowSymLinks

    AllowOverride None
    Require all granted

</Directory>
```

This grants permissions for directory browsing (`Indexes`), symbolic links (`FollowSymLinks`), and sets the access control.

5. ServerName

The `ServerName` directive specifies the hostname and port that the server uses to identify itself. It is crucial for virtual hosting and is typically set near the top of `httpd.conf`:

```
ErrorLog "/var/log/httpd/error_log"

CustomLog "/var/log/httpd/access_log" combined
```

6. ErrorLog and CustomLog

ErrorLog and CustomLog directives define the log files for errors and access respectively:

```
ErrorLog "/var/log/httpd/error_log"

CustomLog "/var/log/httpd/access_log" combined
```

Replace the paths with your desired log file locations.

Restart Apache

After making changes to `httpd.conf`, restart the Apache server to apply the new configurations:

```
systemctl restart httpd
```

Vim Shortcuts

- `i`: Insert mode.
- `:w`: Save changes.
- `:q`: Quit.
- `:wq`: Save and quit.