DIPLOMA WALLAH

(Your One Stop Hub For Diploma Resources)

OPERATING SYSTEM AND ADMINISTRATION

Complete Notes Based on Full Syllabus

- Diploma Engineering

4th Semester

www.diplomawallah.in

Notes prepared by **Sangam**

Hypervisor

Diploma Wallah

Sangam

Unit-02

* Virtualization Technology :-

Virtualization is a technology where the resources of a single computer or server (such as CPU, RAM, Storage.) are divided into separate virtual parts.
This allows multiple system or application to run simultaneously on the same physical machine, as if we have one computer but many virtual computer running inside it.

Types of Virtualization

(1) OS Virtualization :-
OS virtualization is a type of virtualization where the operating system is divided into multiple isolated user environments, allowing multiple virtual instance to run on the same physical machine using a single OS kernel.

Function
(I) User Interface
(II) process management
(III) memory management
(IV) file management
(V) device management

(II) Network Virtualization

Network virtualization is a technology that allows you to combine and manage physical networking hardware (like routers, switches, and firewalls) into a single virtual

Example:- Windows Vm for office work
Linux Vm for web hosting.

(III) Application virtualization:-

It is a technology that allow an application to run without being installed directly on the operating system.

Instead the application is an isolated virtual environment.

→ Run app without direct installation on OS.
→ Isolated virtual container.
→ Best for shared environment, secure usage.

Function

→ No direct installation.
→ Application Isolation.
→ Centralized Management
→ On-demand Access.
→ Faster Deployment
→ Simplified updates.

* Desktop Virtualization

→ Desktop virtualization allow users to access a virtual desktop environment stored on a centralized server instead of using a local desktop operating system.

Function

→ Remote access
→ Security
→ Centralized Management
→ Customization - user get personalized desktop.

* Storage virtualization :-

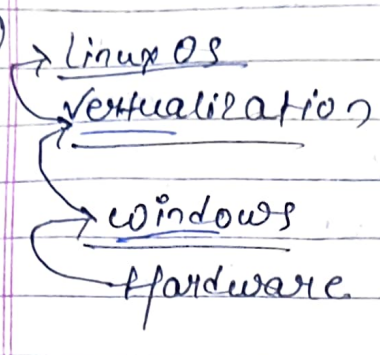
Storage virtualization is a technology that combines multiple physical storage devices (HDDs, SSDs, SAN, NAS) into a single virtual storage unit that appears as one storage system.

* It hides the complexity of physical storage devices and present a single, unified, virtual storage system that is easier to manage and use.

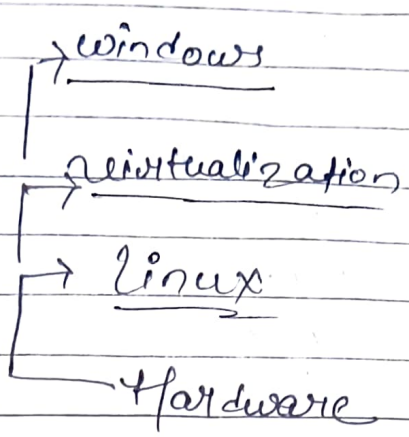
Function :-

→ Pooling of storage :- (combines multiple devices)
→ Simplified Management
→ Data protection
→ Improved Availability.

(I)



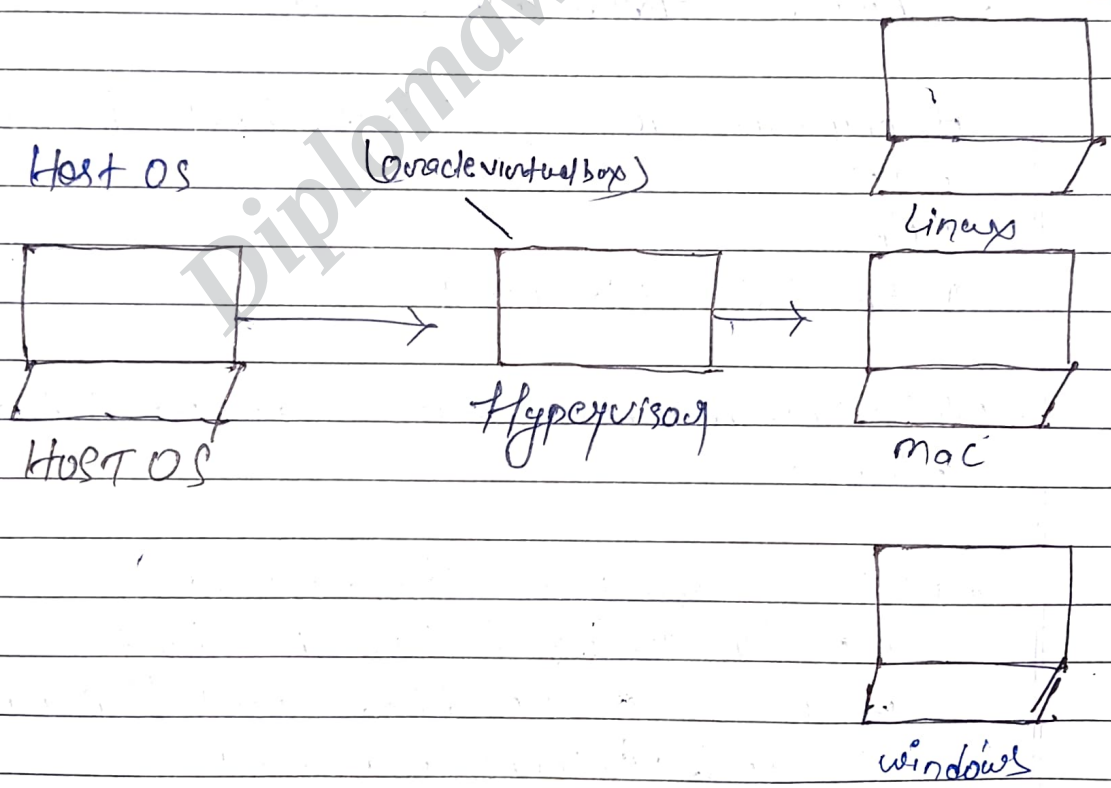
(II)

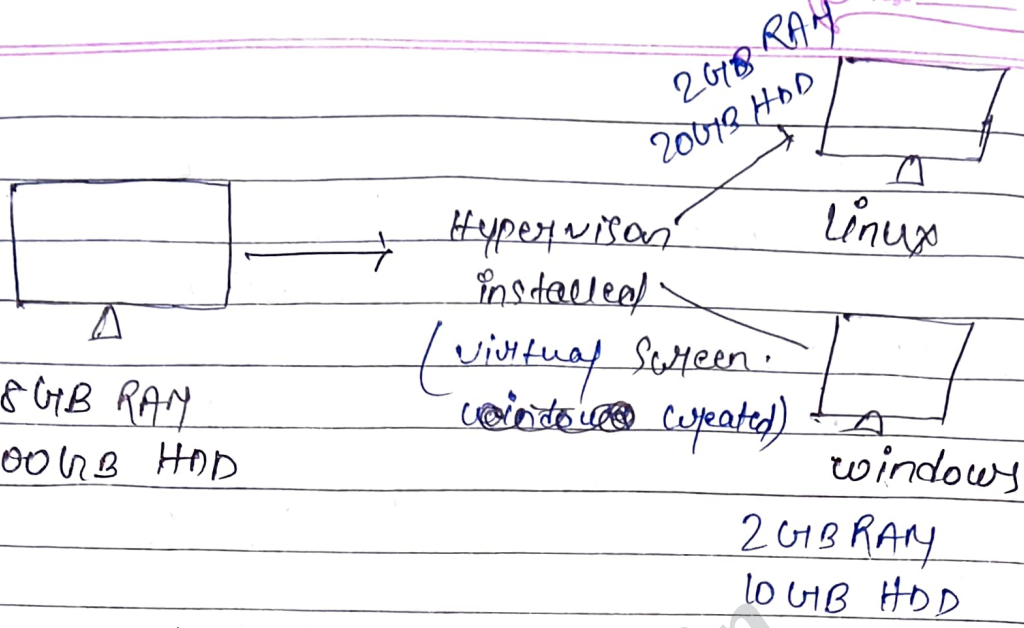


* New virtual machine (OS) create to test our program application.

* work as a layer of i.e Hypervisor/ Software that creates and runs virtual machine

Ex





Working

1. physical Hardware Setup

A physical machine (called host) has resources:-
- CPU
- RAM
- Harddisk, • Network

Note:- The machine must support virtualization mode.

Step 2:- Install a Hypervisor
- A hypervisor installed in host.
- Acts as a middle layer b/w hardware and virtual machine.

You can use:-
- Type 1 (bare-metal): runs directly on hardware
- Type 2: (hosted): runs on an OS (eg virtualBOX).

Step3:- Create virtual Machines (VMs)
- Inside the hypervisor, we can create one or more VMs.
- While creating a VM, we allocate:-
  - RAM (e.g 4GB)
  - CPU core (e.g. 2 cores)
  - Storage (eg 100GB)
  - Network Settings.

Step4:- Install operating System on VMs.
- just like a real PCs. we install an operating system on each VM.
  Eg- window, Linux, macOS.
- Each VM is isolated and run its OS independently.

Step5:- Run VMs concurrently
- We can start and run multiple VMs at the same time.
- All VMs share the physical resources of the host.
- Hypervisor manage and schedule CPU time, RAM etc. for each VM.

Step6:- User Accers and use
- users can log in to each VM and use it like a normal computer.

Step7:- Manage and Monitor Vms.

Advantages (potential)

* Better Resources utilization.
* We don't need new resources to use different OS.
* No risk of any issues with your primary OS.
* Testing any app on different OS.
* Easy Backup & Recovery :— Snapshots and cloning allow fast backup, restore.
* Each VM is separated; if one crashes, others are unaffected.
* Old operating system or apps can run in a VM even if not supported natively.

Disadvantage (Challanges)

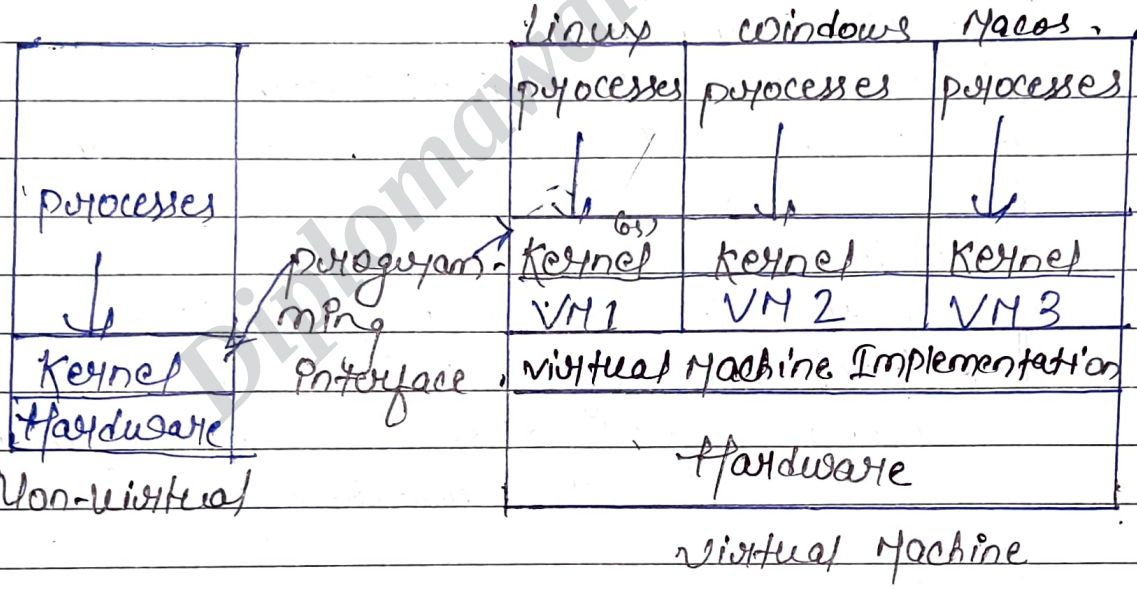
(I) VMs may run slower than physical machines due to shared resources
(II) Requires Skilled IT staff to manage VMs, network and security settings.
(III) Security Risk
(IV) Some older CPUs or systems may not support hardware virtualization.
(V) VMs need full image backup, which can consume more storage.
(VI) If many VMs run at once, they may complete for limited CPU/RAM resources.

Diploma Wallah

Sangam

Virtual Machines

The fundamental idea behind a virtual machine is to abstract the hardware of a single computer (the CPU, memory, disk drivers, network interface card and so forth) into several different execution environments, thereby creating the illusion that each separate execution environment is running its own private computer.



* Implementation
* Virtual Machine Software - Runs in Kernel Mode.
* Virtual machine itself - Runs in user mode.
  ↳ VM1, VM2, VM3
* Just as the physical machine has two modes, however, so must the virtual machine. Must have a virtual user mode and a virtual kernel mode. (Both which run in a physical user mode)

How it work?

→ Install a hypervisor (like VirtualBox) on physical machine.
→ Create a VM in the hypervisor.
→ Allocate some resources
→ Install an operating system (like Linux) in the VM.
→ Now, open the VM and use like a real PC!

Advantage

→ Runs multiple OS on one computer.
→ Test software safely.
→ Backup and restore easily.
→ Ideal for learning, experimenting and development.

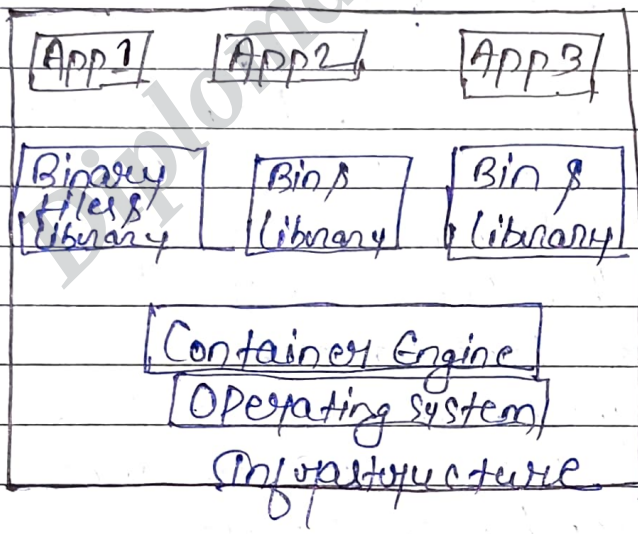
Disadvantage

→ Slightly slower than physical PCs.
→ uses a lot of system resources
→ Need a good setup to run multiple VMs smoothly.

Virtual Container

* A container which is lightweight, portable runtime environment that includes an app and its dependencies.

* Containers run on top of a host OS using a container engine like Docker.
  It doesn't need a full operating system inside it.
* multiple containers can run and installed in the same OS.



Container.

* Main difference b/w virtual machine and container is in container we did not need to create different virtual OS.
  but the OS function will be same in all Application interfaces. (desktop 1/3)

Linux

Linux is a free and open-source os based on unix.

* Kernel is the core part of linux that manages hardware.
* It is open source. anyone can view, change and share it code.
* many user can use the same system.

* Linux Boot process:-

Step1:- power ON the computer
when we power on the computer a small program called BIOS OR UEFI checks if the hardware (RAM, Keyboard, hardisk)
(Uniand unified bootloader)

Step2: Bootloader starts (GRUB)
- After the check, BIOS finds and loads the bootloader.
- In linux, we often use GRUB (GNU GRUB loader).
- If you have multiple OS (like windows + Linux) GRUB lets you choose which one to start.

(GRUB is like a traffic officer deciding which road (os) to take).

Step 3: Linux Kernel is loaded.
- GRUB loads the Linux kernel into memory.
- The Kernel is heart of Linux - it controls all hardware like USB, keyboard, memory etc.

(Kernel is like a manager that runs everything behind the scenes.)

Step 4: Initramfs (Temporary filesystem) is loaded.
- A temporary environment called initramfs (or initrd) is loaded.
- It contains the basic tools the system needs to continue booting.

(Initramfs is like a survival kit with essential tools.)

Step 5:- Systemd or init process starts.
- The Kernel now starts the first process : Systemd (or old System use init).
- It start all necessary services like:- network, disk mount, display, sounds etc.

(Systemd is like a conductor starting the orchestra of services.)

Step 6: Login Screen Appears.
- Now the login screen or terminal appears.
- We enter our username and password to log in.

(This is like gate to your personal workspace.)

Step 7: Desktop or Terminal Starts.

- After login, our desktop (like GNOME, KDE) or command line (shell) opens.
- We are now ready to use Linux.

Flow chart

BIOS/UEFI
(hardware check)
↓
Bootloader (GRUB)
- Choose OS
↓
Load Linux Kernel
↓
Load Initramfs
(Temporary files)
↓
Start systemd / init
(Start services)
↓
Login screen
↓
Desktop / Shell
(Use environment)

Linux Command Line →
command line interface is a text based user interface that lets the user interact with the OS by typing command, instead of using a mouse or graphical windows.

Example
- Terminal in Linux
- Command prompt in windows
- SSH (used for remote Linux control)

* Command line Interpreter
A Command line Interpreter is a software program that acts as a middleman b/w the user and the kernel.
It reads user input, interprets it, and then executes the appropriate system calls or programs.

(i) It read the command from user.
(ii) parse and understand what the command means.
(iii) Execute the command using functions.

→ Comman Linux command

* bash	most common shell (Bourne Again shell)
* sh	original unix shell (Bourne Shell)
* zsh	Z shell, modern with more features
* fish	user-friendly interactive shell
* dash	fast, lightweight interpreter used in scripts.

Shell

A shell is a command line interpreter that provides a user interface for accessing the services of the OS.

The shell is also run script, automate task and manage input/output.

function

* Input handling
* Interpretation
* Execution
* Output display
* Scripting (executes shell scripts)

popular shell in Linux

bash	Default on most system
sh	Older Unix shell
zsh	Advance shell with plugin and themes.
fish	Very user-friendly with suggestion.
csh	shell with C-style syntax.

CLI over GUI

* The command line interface (CLI) is the environment or interfaces where user input textual content.
* The CLI (such as bash, zsh, sh) is the program that reads, interprets and executes these commands.

CLI - front-end
Interpreter - back-end

* Command line interface (where we write) and the interpreter runs the command entered by user.

Types of users

Linux is a multiuser operating system where many user can use the system at the same time.

There are mainly three categories →

① Superuser (root user)
The Superuser is a special user account with UID 0 and unrestricted access to all system resources.

Characteristics:
- Username: root
- UID: 0
- Can read/write/execute all files regardless of permission.
- Can modify Kernel settings, user accounts, and system configurations.
- Can execute all administrative commands.

Key file access
- /etc/shadow (password hashes)
- /etc/passwd (user database)
- /etc/sudoers (sudo privileges)

Examples
sudo visudo // Edit sudo permission
sudo useradd dev // Create a user (require root privileges).

(II) Normal user

user created by administrator or during installation with UID > 1000, intended for interactive use.

Characteristics

- UID: typically start from 1000 and increases.
- Limited access: can only access files they own.
- Cannot access or modify system-critical files.
- Can use sudo (if permitted) to run administrative commands.

- Restriction
  - No access to /etc, /root, or /usr/bin
  - Cannot install software unless added to the sudo group.

Example:

bash

```
whoami                  // Show current user
cd ~/Download           // Navigate to user home directory.
```

(iii) Linux user manual

The linux user manual is a built in documentation system that provides detailed help pages for nearly every command, file, system, call and configuration file.

We access it using the man command

Command Syntax:—

man [section] command_name

man → open manual
section → optional number (to specify section)
command_name → the linux command or topic.

We use it because:—

→ To learn how command works
→ To view all available options/switches for command.
→ To see file format definations, system calls or kernel functions.

✗ The linux user manual is a system of online help pages accessed using the man command, categorized into section covering commands, configuration file system calls, and administrative tools.