2 implementiations of system call

What is a System Call?

A system call is a way for a program to request a service from the operating system's kernel. These services can include tasks like reading from a file, writing to a file, creating or terminating processes, and managing memory.

Since user applications cannot directly access kernel-level operations for security and stability reasons, system calls act as a controlled interface between user programs and the operating system.

Common Examples of System Calls:

Read () – to read data from a file

Write () – to write data to a file

Fork () – to create a new process

Exec () – to run a new program

Exit () – to terminate a process

For Ubuntu server we use shutdown system for shutting down a system

Implementation of shutdown () system call in Ubuntu server

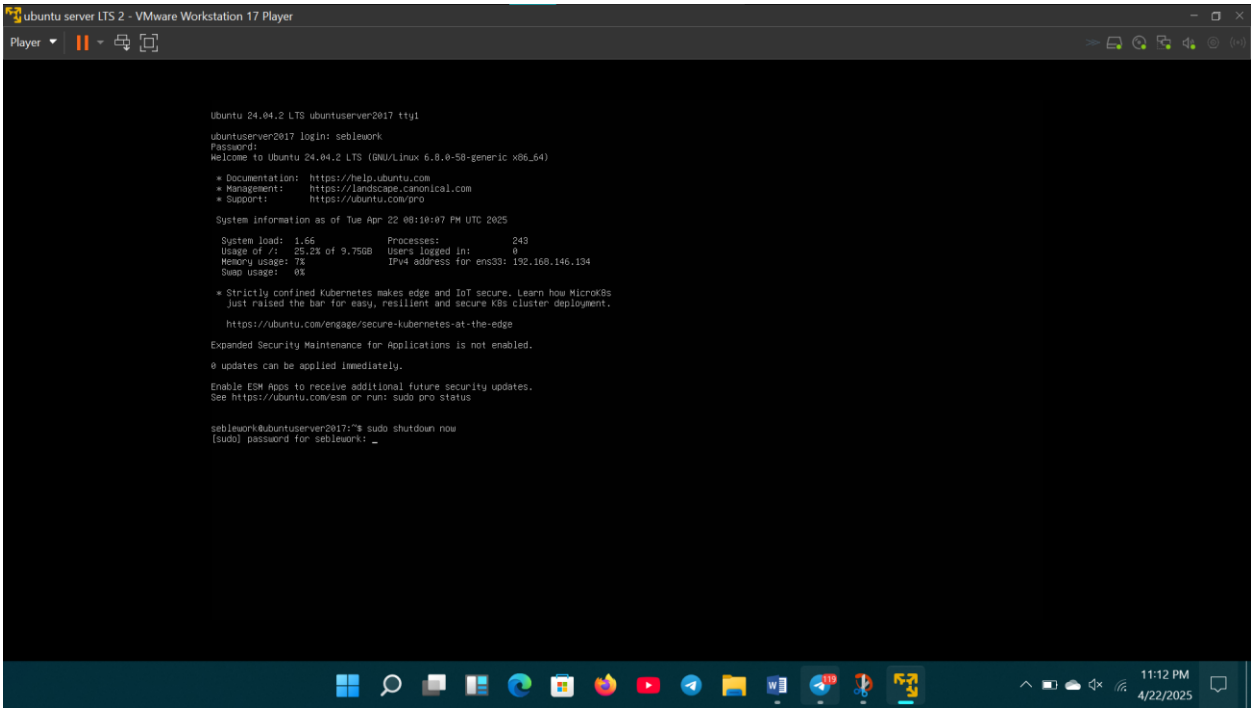
For shutting down a system, use built-in commands like shutdown, halt, or poweroff directly from the terminal without writing a custom program.

Example of Using Terminal Commands

- sudo shutdown now

• Purpose: This command is used to immediately shut down the system.

• Functionality: It sends a signal to all running processes to terminate and then powers off the machine. The sudo prefix indicates that the command requires superuser (administrative) privileges to execute.

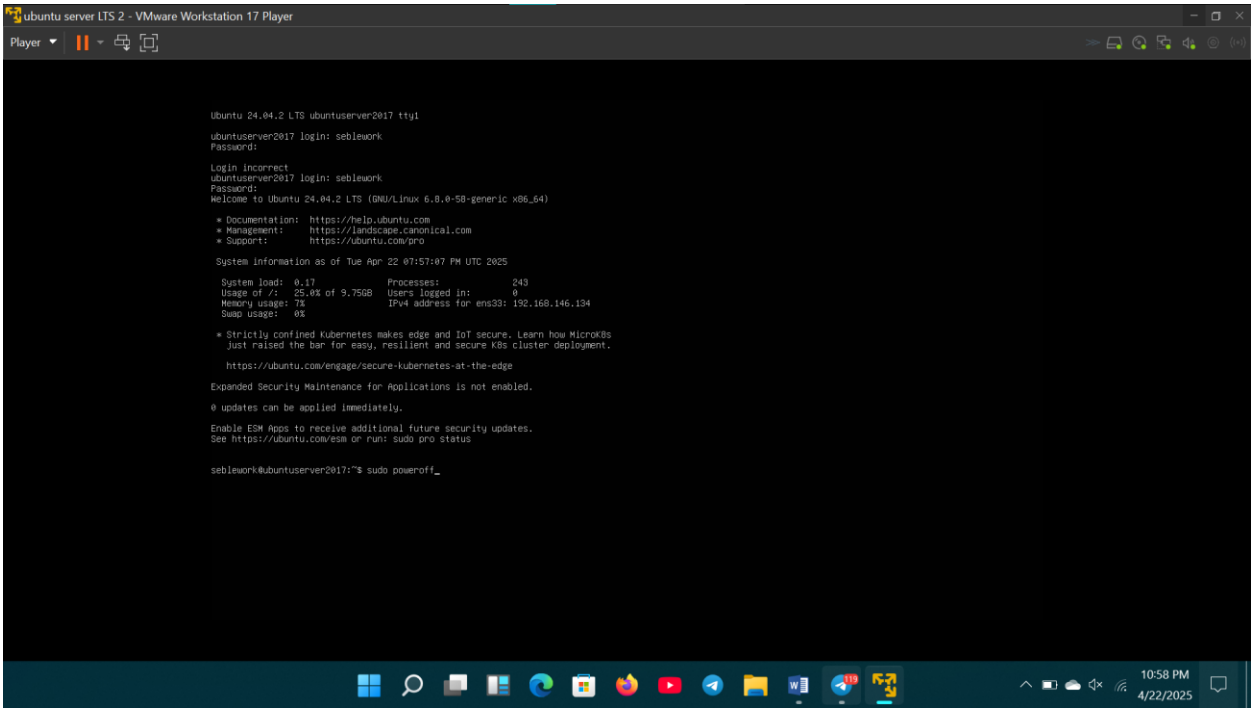


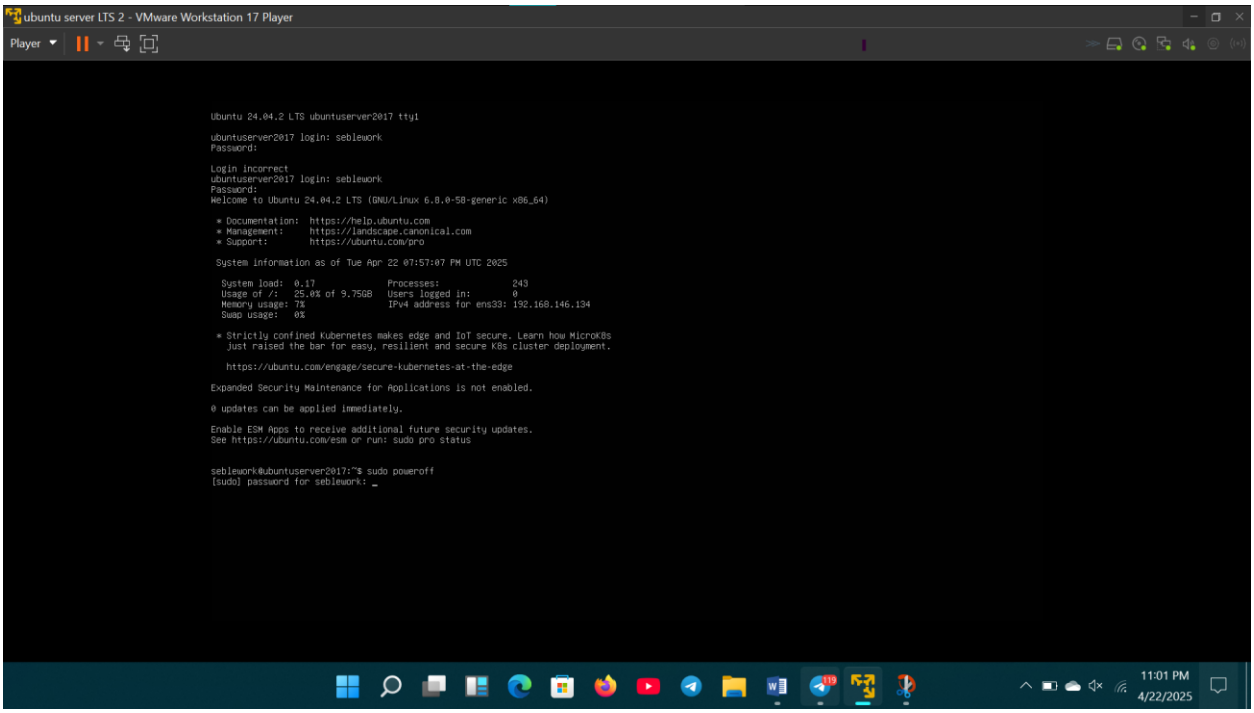
or

- ❖ sudo poweroff

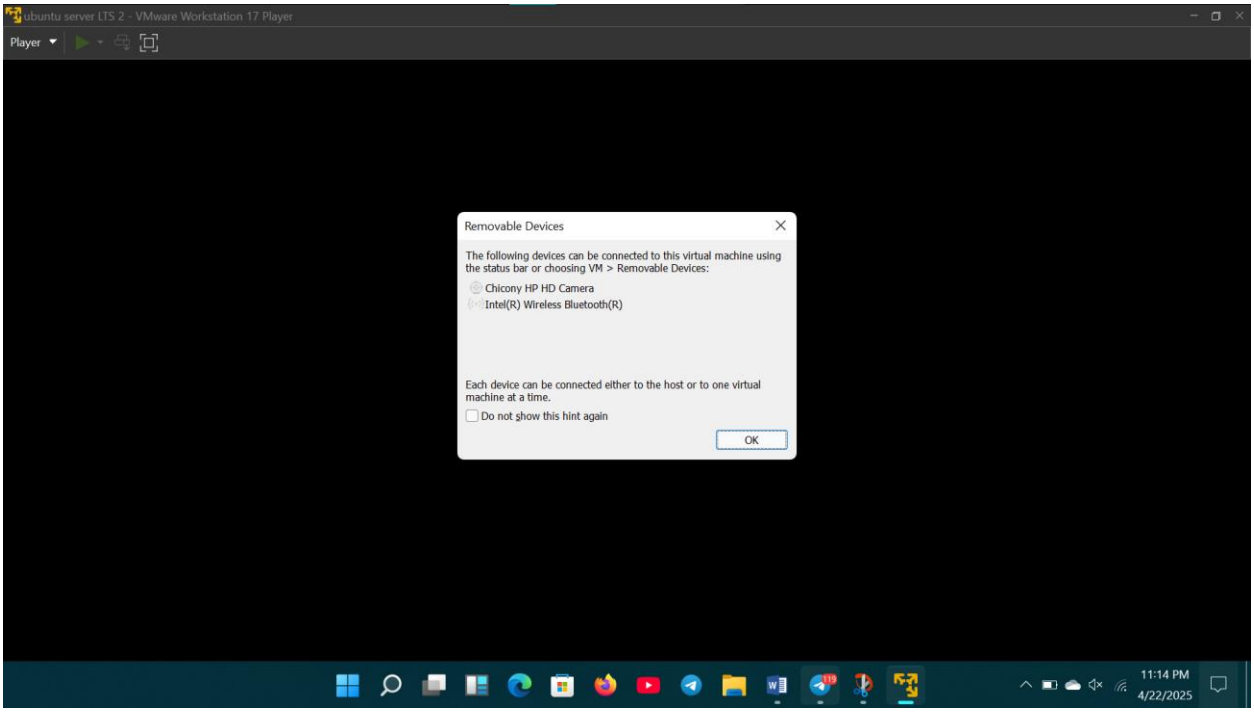
• Purpose: This command is also used to turn off the system.

• Functionality: Similar to shutdown, it immediately powers off the machine, but it may bypass some of the graceful shutdown procedures. It effectively stops all processes and powers down the hardware directly.





These commands will also safely shut down your Ubuntu server and both results to shutdown system



3. Virtualization in modern operating system

3.1Introduction

Virtualization is technology that allows to create virtual representations of servers, storage, networks, and other physical machines. Virtual software mimics the functions of physical hardware to run multiple virtual machines simultaneously on a single physical machine. Businesses use virtualization to use their hardware resources efficiently and get greater returns from their investment. It also powers cloud computing services that help organizations manage infrastructure more efficiently.

Virtualization is a technique that allows multiple operating systems and applications to run simultaneously on a single physical machine, each within its own virtual environment. This technology forms the backbone of cloud computing, data centers, and development platforms, helping organizations optimize hardware usage, reduce costs, and improve system agility. Modern operating systems are no longer limited to managing resources on a single physical machine; instead, they play a crucial role in enabling powerful and abstracted computing environments through virtualization.

3.2Understanding virtualization

Virtualization is the creation of a virtual version of an actual piece of technology, such as an operating system (OS), a server, a storage device or a network resource.

Virtualization uses software that simulates hardware functionality to create a virtual system. This practice lets IT organizations run multiple OS, more than one virtual system and various applications on a single server. The benefits of virtualization include greater efficiency and economies of scale.

OS virtualization uses software that enables a piece of hardware to run multiple operating system images at the same time. The technology got its start on mainframes decades ago to save on expensive processing power.

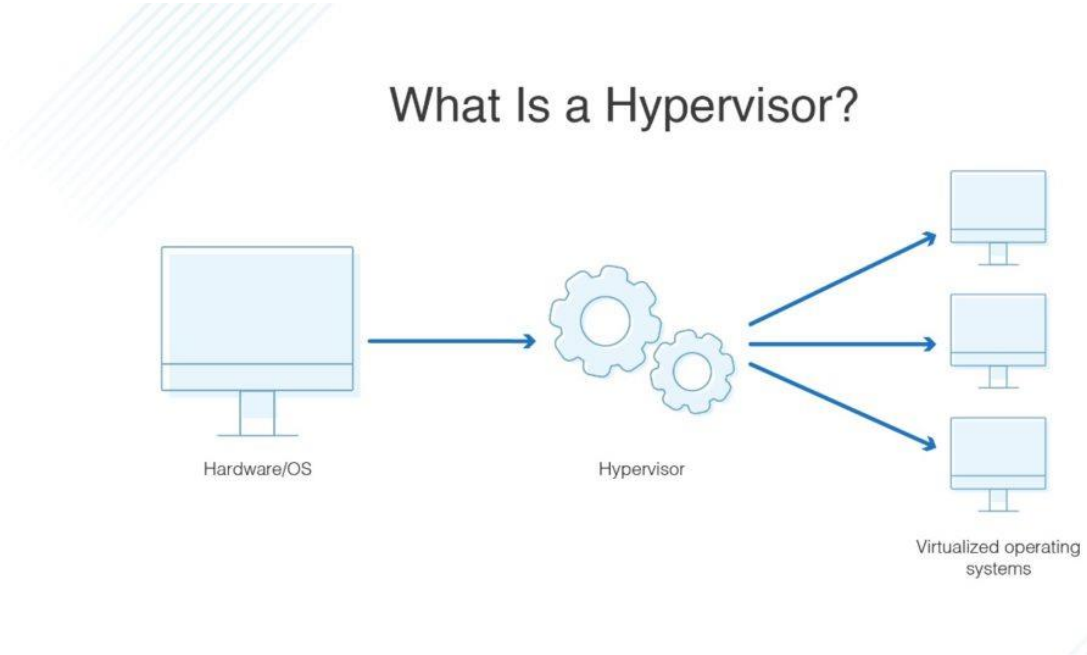
Virtualization is a process that allows a computer to share its hardware resources with multiple digitally separated environments. Each virtualized environment runs within its allocated resources, such as memory, processing power, and storage. With virtualization, user can switch between different operating systems on the same server without rebooting.

Virtual machines and hypervisors are two important concepts in virtualization.

- Virtual machine

A virtual machine is a software-defined computer that runs on a physical computer with a separate operating system and computing resources. The physical computer is called the host machine and virtual machines are guest machines. Multiple virtual machines can run on a single physical machine. Virtual machines are abstracted from the computer hardware by a hypervisor.

- Hypervisor

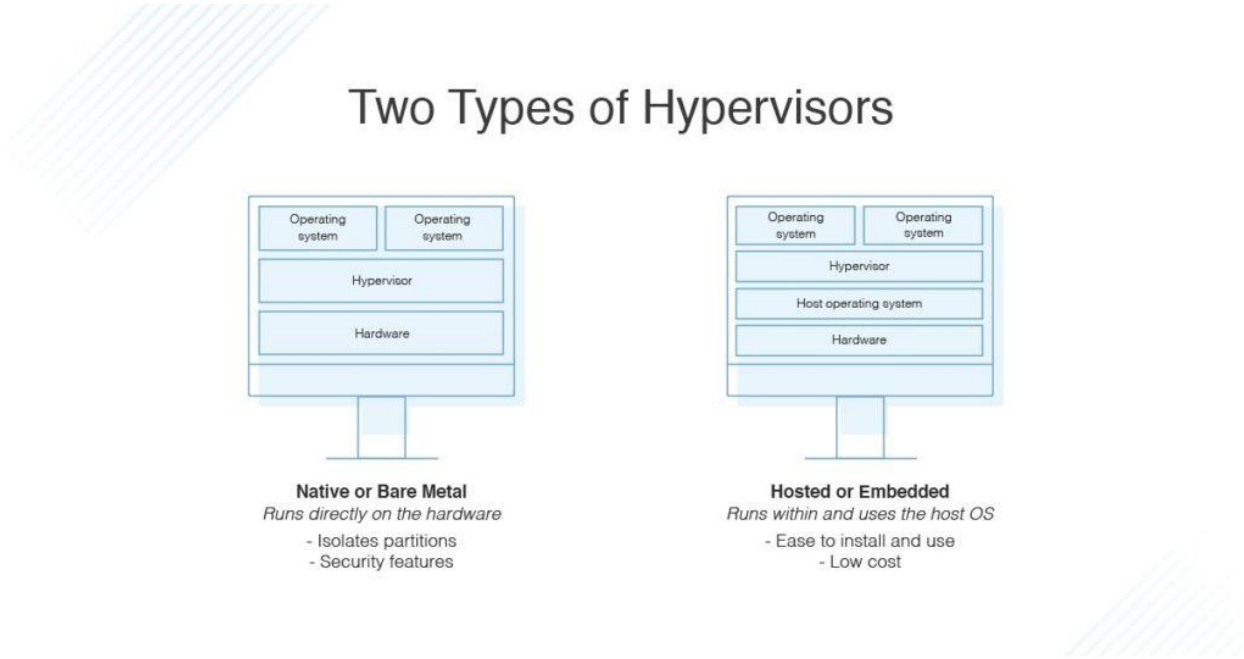


The hypervisor is a software component that manages multiple virtual machines in a computer. It ensures that each virtual machine gets the allocated resources and does not interfere with the

operation of other virtual machines. Hypervisors were created in 1965 to work with the IBM RPQ for the IBM 360/65. They were originally designed to test sharing systems between virtual machines and looking at new hardware concepts without jeopardizing the main production system. Now, hypervisors are commonly used to allocate physical hardware resources to virtual machines, which are known as “guests” on the host machine.

Hypervisors are used for many different tasks, including cloud computing, server management, and simply running programs y compatible with an operating system you don’t have. You can use a hypervisor to run processes and operating systems on virtual machines, totally separate from your main system.

There are two types of hypervisors.



✓ Type 1 hypervisor

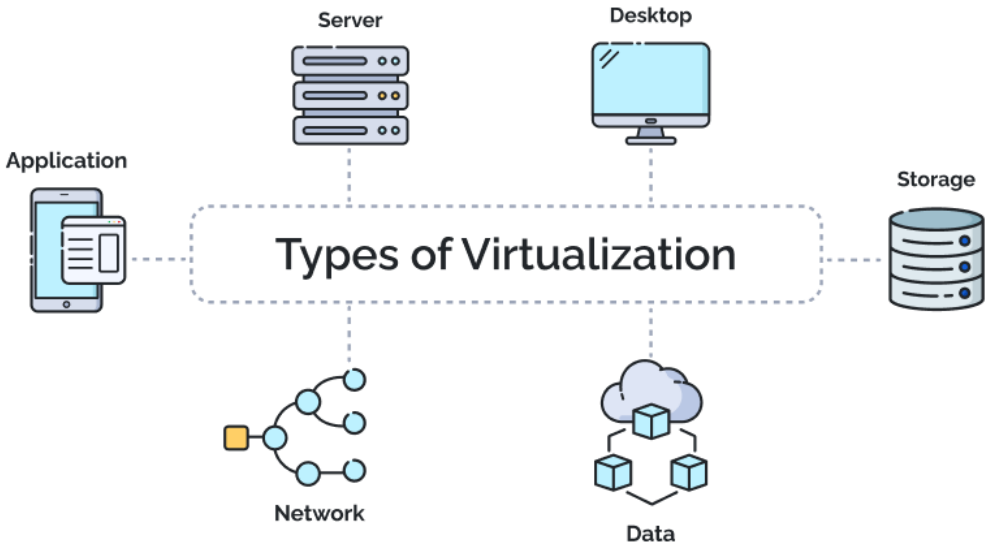
A type 1 hypervisor, or bare-metal hypervisor, is a hypervisor program installed directly on the computer’s hardware instead of the operating system. Therefore, type 1 hypervisors have better performance and are commonly used by enterprise applications. KVM (kernel based virtual machine) uses the type 1 hypervisor to host multiple virtual machines on the Linux operating system.

✓ Type 2 hypervisor

A Type 2 hypervisor is a virtual machine (VM) manager that is installed as a software application on an existing operating system (OS). It is installed on the host OS and does not directly interact with the underlying host machine's hardware. Rather, the interactions happen through the host OS. For this reason, a Type 2 hypervisor is also called a *hosted hypervisor*.

VMware and Hyper-V are two key examples of hypervisor, with VMware owned by Dell and Hyper-V created by Microsoft. VMware software is made for cloud computing and virtualization, and it can install a hypervisor on the physical servers to allow multiple virtual machines to run at the same time. Hyper-V does the same thing, but you can also virtualize servers. Both are bare metal (native) hypervisors. Oracle VM VirtualBox is a hosted hypervisor.

3.3 Types of virtualization



- Server virtualization

Server virtualization is a process that partitions a physical server into multiple virtual servers. It is an efficient and cost-effective way to use server resources and deploy IT services in an organization. Without server virtualization, physical servers use only a small amount of their processing capacities, which leave devices idle.

- Storage virtualization

Storage virtualization combines the functions of physical storage devices such as network attached storage (NAS) and storage area network (SAN). You can pool the storage hardware in your data center, even if it is from different vendors or of different types. Storage virtualization uses all your physical data storage and creates a large unit of virtual storage that you can assign and control by using management software. IT administrators can streamline storage activities, such as archiving, backup, and recovery, because they can combine multiple network storage devices virtually into a single storage device.

- Network virtualization

Any computer network has hardware elements such as switches, routers, and firewalls. An organization with offices in multiple geographic locations can have several different network technologies working together to create its enterprise network. Network virtualization is a process that combines all of these network resources to centralize administrative tasks. Administrators can adjust and control these elements virtually without touching the physical components, which greatly simplifies network management.

The following are two approaches to network virtualization.

Software-defined networking

Software-defined networking (SDN) controls traffic routing by taking over routing management from data routing in the physical environment. For example, you can program your system to prioritize your video call traffic over application traffic to ensure consistent call quality in all online meetings.

Network function virtualization

Network function virtualization technology combines the functions of network appliances, such as firewalls, load balancers, and traffic analyzers that work together, to improve network performance.

- Data virtualization

Modern organizations collect data from several sources and store it in different formats. They might also store data in different places, such as in a cloud infrastructure and an on-premises data center. Data virtualization creates a software layer between this data and the applications that need it. Data virtualization tools process an application's data request and return results in a suitable format. Thus, organizations use data virtualization solutions to increase flexibility for data integration and support cross-functional data analysis.

- Application virtualization

Application virtualization pulls out the functions of applications to run on operating systems other than the operating systems for which they were designed. For example, users can run a Microsoft Windows application on a Linux machine without changing the machine configuration. To achieve application virtualization, follow these practices:

Application streaming – Users stream the application from a remote server, so it runs only on the end user's device when needed.

Server-based application virtualization – Users can access the remote application from their browser or client interface without installing it.

Local application virtualization – The application code is shipped with its own environment to run on all operating systems without changes.

- Desktop virtualization

With desktop virtualization, an admin can deploy simulated desktop environments on a hosted, centralized or remote server. So users can access their desktops on any workstation or device. Furthermore, it makes it easy for admins to perform updates, security checks and mass configurations on all virtual desktops.

The following are types of desktop virtualization based on whether the operating system instance is local or remote.

- **Remote desktop virtualization**

Remote desktop virtualization is a common use of virtualization that operates in a server computing environment. This allows users to run operating systems and applications from a server inside a data center while all user interactions take place on a client device such as a laptop, thin client, or smartphone. This type of virtualization gives IT more centralized control over applications and desktops, and can maximize an organization's investment in hardware through remote access to shared computing resources.

- **Local desktop virtualization**

Local desktop virtualization means the operating system runs on a client device using hardware virtualization, and all processing and workloads occur on local hardware. This type of desktop virtualization works well when users do not need a continuous network connection and can meet application computing requirements with local system resources. However, because this requires processing to be done locally you cannot use local desktop virtualization to share virtual machines (VMs) or resources across a network to thin clients or mobile devices.

A popular type of desktop virtualization is virtual desktop infrastructure (VDI). VDI is a variant of the client-server model of desktop virtualization, which uses host-based virtual machines (VMs) to deliver persistent and nonpersistent virtual desktops to all kinds of connected devices. With a persistent virtual desktop, each user has a unique desktop image they can customize with apps and data, knowing it will be saved for future use. A nonpersistent virtual desktop infrastructure allows users to access a virtual desktop from an identical pool when they need it; once the user logs out of a nonpersistent VDI, it reverts to its unaltered state. Some of the advantages of virtual desktop infrastructure are improved security and centralized desktop management across an organization.

3.4 Important of Virtualization (Why)

Virtualization plays a pivotal role in the way modern computing environments are structured, managed, and utilized. As computing needs grow more complex, the ability to isolate, consolidate, and optimize workloads has become essential. Virtualization addresses these challenges by enabling efficient resource use, enhancing security, and improving system flexibility.

- Efficient Resource Utilization

  Traditionally, many physical servers operate far below their maximum capacity, leading to underutilized hardware. Virtualization allows multiple virtual machines to run on a single physical machine, each with its own operating system and applications. This enables organizations to maximize hardware usage, reducing waste and improving overall system efficiency.
- Cost Savings

By consolidating workloads and reducing the number of physical servers, organizations can significantly cut:
Hardware acquisition costs
Energy consumption (power and cooling)
Physical space requirements in data centers
Additionally, virtualization reduces maintenance and operational costs by simplifying system management and updates.

- Flexibility and Portability
  Virtual machines are encapsulated into files, which makes them easy to:
  Move between physical hosts
  Clone or replicate
  Back up or restore
  This flexibility is crucial in dynamic environments like cloud computing or DevOps pipelines, where systems need to scale quickly or be redeployed frequently.
- Enhanced Security and Isolation
  Each VM or container runs in an isolated environment. This means:
  Faults or security breaches in one VM do not affect others.
  Developers can safely test code or run untrusted applications in isolated sandboxes.
  This level of separation adds a critical layer of security, especially in multi-tenant environments like cloud platforms.
- Simplified System Management and Maintenance
  Virtualization enables centralized control of infrastructure, allowing administrators to:
  Automate system provisioning and updates
  Roll out patches more efficiently
  Use templates for consistent deployments
  Tools like VMware vSphere or Microsoft System Center enhance visibility and control over virtual environments.
- Scalability and Elasticity
  Virtualized environments can scale resources up or down based on demand. For instance:
  Adding more RAM or CPUs to a VM can be done with minimal downtime.
  Workloads can be balanced across multiple hosts using clustering or live migration.
  This elasticity is especially important for cloud providers and large-scale applications.
- Business Continuity and Disaster Recovery
  Virtual machines can be easily backed up, replicated, or restored in the event of hardware failure or disasters. Many platforms support:
  Live migration (moving VMs between hosts with no downtime)
  Snapshots (restoring VMs to a previous state)
  Geographic replication for off-site disaster recovery
  This improves uptime and ensures business continuity.
- Support for Legacy Systems
  Virtualization allows older operating systems or applications to run on modern hardware, preserving access to legacy systems without needing to maintain outdated hardware.
  This is particularly useful for industries relying on software that is no longer actively developed.
- Development and Testing
  Developers benefit immensely from virtualization by:

Creating multiple test environments quickly
Running different OS configurations side-by-side
Isolating bugs and testing without affecting production systems
Tools like Vagrant and Docker have revolutionized the way software is built, tested, and deployed.

Virtualization is not just a way to improve hardware efficiency—it is a comprehensive approach that enhances system performance, security, scalability, and manageability. These benefits make virtualization a cornerstone of modern IT infrastructure, from personal computing to enterprise-level cloud environments..

3.5 Implementation of virtualization (HOW)

The Five Levels of Implementing Virtualization

1. Instruction Set Architecture Level (ISA)
2. Hardware Abstraction Level (HAL)
3. Operating System Level
4. Library Level
5. Application Level

Instruction Set Architecture Level (ISA)

virtualization works through an ISA emulation. This is helpful to run heaps of legacy code which was originally written for different hardware configurations.

Hardware Abstraction Level (HAL)

As the name suggests, this level helps perform virtualization at the hardware level. It uses a bare hypervisor for its functioning.this level helps form the virtual machine and manages the hardware through virtualization.It enables virtualization of each hardware component such as I/O devices, processors, memory, etc.This way multiple users can use the same hardware with numerous instances of virtualization at the same time.IBM had first implemented this on the IBM VM/370 back in 1960. It is more usable for cloud-based infrastructure. Thus, it is no surprise that currently, Xen hypervisors are using HAL to run Linux and other OS on x86 based machines.

Operating System Level

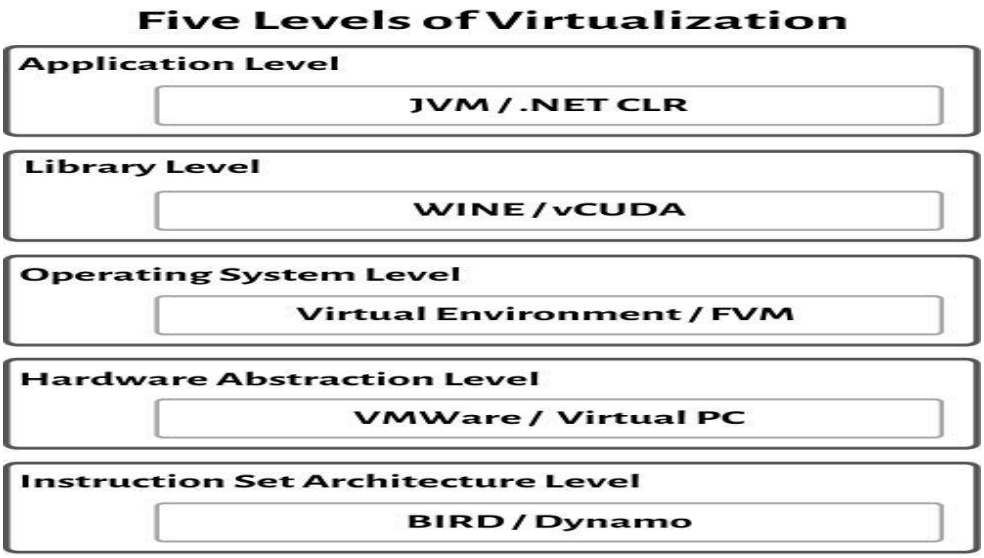
At the operating system level, the virtualization model creates an abstract layer between the applications and the OS.It is like an isolated container on the physical server and operating system that utilizes hardware and software. Each of these containers functions like servers.When the number of users is high, and no one is willing to share hardware, this level of virtualization comes in handy.Here, every user gets their own virtual environment with dedicated virtual hardware resources. This way, no conflicts arise.

Library Level

OS system calls are lengthy and cumbersome. Which is why applications opt for APIs from user-level libraries. Most of the APIs provided by systems are rather well documented. Hence, library level virtualization is preferred in such scenarios. Library interfacing virtualization is made possible by API hooks. These API hooks control the communication link from the system to the applications.Some tools available today, such as vCUDA and WINE, have successfully demonstrated this technique.

Application Level

Application-level virtualization comes handy when you wish to virtualize only an application. It does not virtualize an entire platform or environment.On an operating system, applications work as one process. Hence it is also known as process-level virtualization.It is generally useful when running virtual machines with high-level languages. Here, the application sits on top of the virtualization layer, which is above the application program.The application program is, in turn, residing in the operating system.Programs written in high-level languages and compiled for an application-level virtual machine can run fluently here.



Virtualization is done by using a special layer of software called a hypervisor (or Virtual Machine Monitor) that separates physical hardware from the virtual machines (VMs) or environments running on it. Implementation follows these step

1. **Installing a Hypervisor**

A hypervisor is installed either directly on hardware (Type 1) or on top of an existing operating system (Type 2).

Type 1 (bare-metal): VMware ESXi, Microsoft Hyper-V, KVM.
Type 2 (hosted): Oracle VirtualBox, VMware Workstation.

2. **Creating Virtual Machines**

The hypervisor allows you to create multiple virtual machines, each with its own virtual CPU, memory, disk, and network interfaces.
These VMs run independently with their own operating systems and applications.

3. Allocating Resources

Physical hardware resources (CPU, RAM, storage, etc.) are split and allocated to VMs by the hypervisor.
VMs can share resources and the hypervisor ensures that they do not interfere with each other.

4. **Running Applications**

Once a VM is powered on, it operates like a real computer.
You can install an OS (like Linux or Windows) and run applications inside it.

5. Management and Monitoring

Virtualization platforms provide management tools (e.g., VMware vCenter, Hyper-V Manager) for:

5 **Starting/stopping VMs**

- Taking snapshots and backups
- Monitoring performance
- Migrating VMs between hosts (live migration)

6. **Virtualization Technologies**

Hardware-assisted virtualization: Modern CPUs (Intel VT-x, AMD-V) include features that support virtualization more efficiently.
Container virtualization: Uses OS-level features (like Linux namespaces and cgroups) to create lightweight, isolated environments.

3.6 Advantages and Disadvantages of Virtualization

Advantages:

- Better resource utilization
- Cost savings
- Faster provisioning
- Simplified management
- Enhanced disaster recovery
- Greater flexibility and scalability

Disadvantages:

- Performance overhead
- Licensing costs
- Complex management
- Security vulnerabilities
- Not suitable for all workloads

Conclusion

virtualization represents a major advancement in how computing resources are managed, allocated, and secured. It abstracts the hardware layer and allows multiple operating systems or instances of the same OS to run simultaneously on a single physical machine, each in its own isolated environment.

Virtualization is deeply rooted in core OS concepts such as process management, memory management, file systems, and security. Through technologies like hypervisors (Type 1 and Type 2), the OS gains the ability to simulate hardware resources, control execution environments, and enforce strict isolation between virtual machines (VMs). This is particularly useful for creating safe testing environments, efficient server utilization, and fault isolation.

virtualization supports key OS design goals such as:

Efficiency: By sharing CPU, memory, and storage resources across virtual environments.

Scalability: Allowing systems to expand quickly by spinning up new virtual instances.

Security: Enforcing strong isolation between VMs, reducing the risk of cross-system attacks.

Portability: Abstracting the underlying hardware, making OS instances easier to migrate or replicate.

Virtualization also lays the groundwork for modern innovations such as cloud computing, containers (like Docker), and platform-as-a-service (PaaS) models, all of which are built on foundational OS mechanisms.

In conclusion, virtualization is not only a practical solution for efficient resource use—it is also a critical concept in the evolution and understanding of operating systems, bridging traditional OS theory with modern infrastructure practice.

References

- IEEE Std 1003.1™-2017 (POSIX). Retrieved from https://standards.ieee.orgUbuntu.com
- GeeksforGeeks. (n.d.). Introduction to Ubuntu Operating System. Retrieved from https://www.geeksforgeeks.org/introduction-to-ubuntu-operating-system/
- Canonical Ltd. (n.d.). Ubuntu Desktop Documentation. Retrieved from https://ubuntu.com
- GeeksforGeeks. (n.d.). Virtualization in Operating System. Retrieved from https://www.geeksforgeeks.org/virtualization-in-operating-system/