PURBANCHAL UNIVERSITY

SCHOOL OF ENGINEERING (PUSOE)

Operating System

Assignment 1

Submitted by: **Barsha Karn**

Roll no : 13 'A'

Faculty: Computer Engineering

Batch Year : 2020

Year/ Semester: 3rd/5th

College: PUSOE, Brt

Submitted to:

Er. Deeoranjan Dongol sir

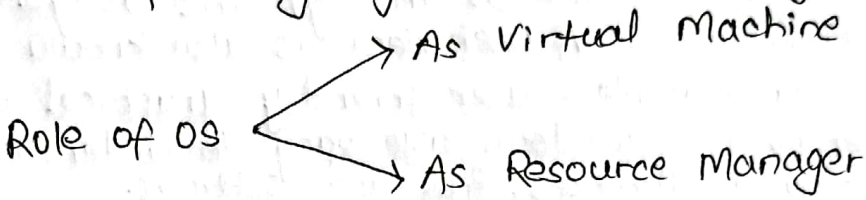
Assignment - 01

Question 1 - What is an operating system? Explain the roles of operating system.

Ans: An operating system (OS) is software that acts as an intermediary between computer hardware and user applic-ations. It manages and controls the computer's resources, proriding an environment in which applications can run efficiently.

In other words, OS is the interface between the application program and hardware level which controls the hardware and makes the hardware usable.
Examples of OS are: Windows, MacOS, Linux, Android, etc.

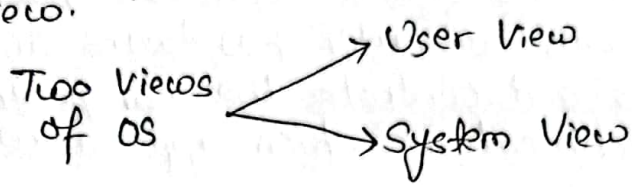
The roles of an operating system can be categorized in two ways:



As Virtual Machine or extended Machine: An OS in the context of VM manges the resources and provides an envir-onment for multiple virtualized instances of an OS to run on a single physical machine. It abstracts hardware resources and allows these instances, called virtual machines to run independently. For example: VMware and VirtualBox are virtualization platforms that enable us to create and run multiple VMs on a single physical computer. The OS runn-ing on the physical machine is responsible for managing hardware resources like CPU, memory and storage, while each VMs runs its own guest OS, unaware of the presence of other VMs.

As Resource Manager: The operating system (OS) serves as a resource manager by efficiently allocating and controlling the system's hardware and software resources. It ensures that various programs and processes run smoothly without interfe-ring with each other. For example, when we run multiple applications on our computer, the OS manages the CPU, memory, disk space and other resources to ensure each program gets its fair share, preventing one program from monopolizing resources and slowing down others.

Question 2- Highlight on views of an operating system.
Ans:- The operating system may be observed from the viewpoint of the user or the system. It is known as the user view and the system view.



User View- The user view depends on the system interface that is used by the users. Some systems are designed for a single user to monopolize the resources to maximize the user's task. In these cases, the OS is designed primarily for ease of use, with little emphasis on quality and none on resource utilization. The user viewpoint focuses on how the user interacts with the OS through the usage of various application programs. A user view is also known as EXTERNAL SCHEMA. For example - user friendly graphical interface such as Icons, Menus, and windows are easy to understand and navigate for accessing applications, files and settings.

System View: The OS may also be viewed as system view where the OS is responsible for managing hardware resources and allocating them to programs and users to ensure maximum performance. From system point view, OS is the program involved with the hardware. OS is allocator which allocates memory resources among various processes. It controls the sharing of resources among programs. It is the program that runs all the time in the system in the form of Kernel. It controls application programs that are not part of Kernel. For eg: When a user A is running a complex data analysis program and User B is editing a document, the OS ensures that both tasks get fair share of CPU so that one user's task doesn't hog all the processing power.

Question 3- Differentiate between time sharing and real time operating system.

Ans:

Time Sharing OS	Real Time OS
① In time sharing OS, quick respo-nse is emphasized for a request.	① In Real time OS, computa-tion tasks are emphasized before its nominative point.
② Switching method / function is available.	② Switching method is not available.
③ Any modification in the program can be possible.	③ Modification doesn't take place.
④ Computer resources are shared to external.	④ Computer resources are not shared to the external.
⑤ It deals with more than processes or applications sim-ultaneously.	⑤ It deals with only one process or application at a time.
⑥ The response is provided to the user within a second.	⑥ The response is provided to the user within time constraint.
⑦ High priority tasks can be preempted by lower priority tasks, making it impossible to guarantee a response time for your critical applications.	⑦ It gives users the ability to prioritize tasks so that the most critical task can always take control of the process when needed.
⑧ Example of time sharing OS include Linux.	⑧ Example of real time OS include: Free RTOS.

Question 4 - Explain virtualization and it's importance.

Ans:- Virtualization is a technology that enables the creation of virtual versions of computer hardware, software, storage devices or network resources. It allows multiple instances of these resources to run independently on a single physical machine or be distributed accross multiple machines. Virtualization provides several benefits including efficient resource utilization, isolation, flexibility and ease of management.

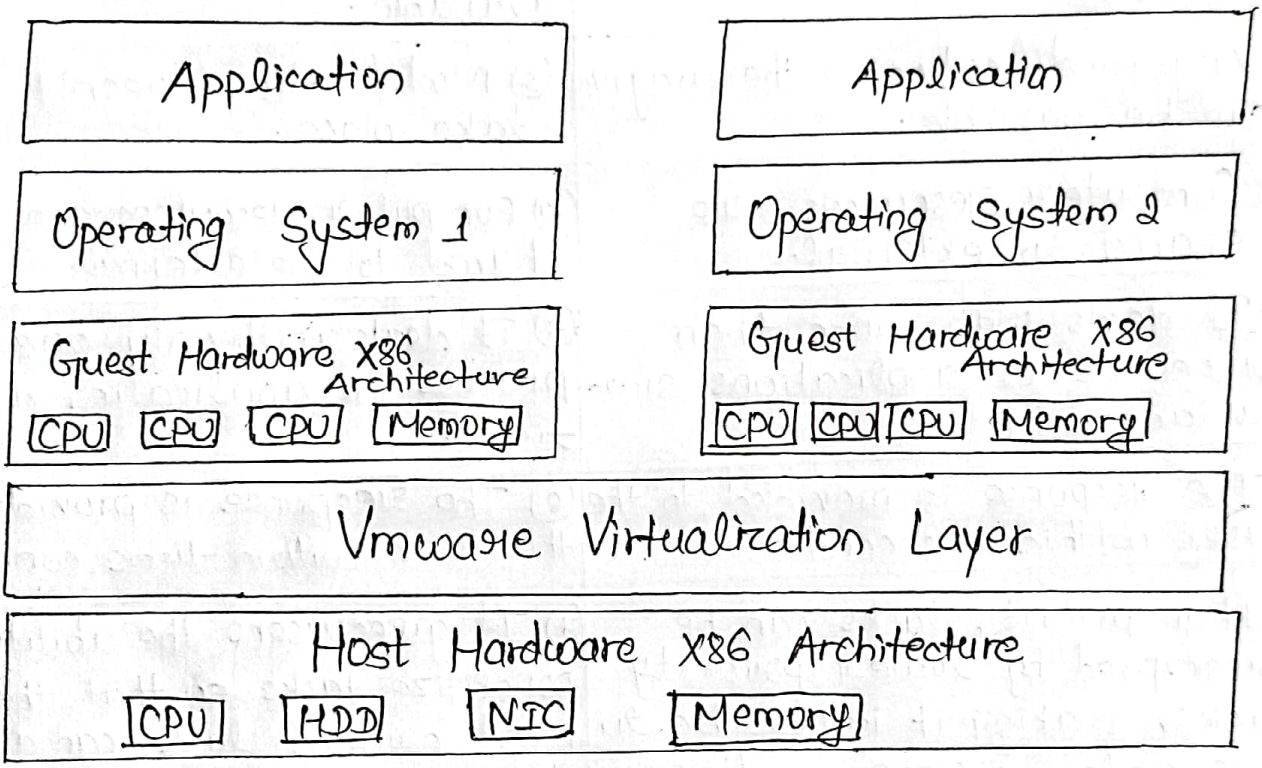


Fig: Virtualization System

The importance of virtualization are:

1. Virtualization maximizes hardware efficiency by running multiple virtual instances one one physical machine.
2. Virtualization provides strong separation between virtual instances, preventing conflicts and security breaches.
3. Virtualization reduces the number of physical servers needed, saving space, power and hardware costs.
4. Virtual instances can be easily created, resized and moved to adapt to changing demands
5. Virtualization simplifies disaster recovery with easily backed up and restored virtual instances.

⑥ Developers use isolated environments to test software changes without affecting production.
⑦ Virtualization allows running older software on modern hardware and software environments.
⑧ Cloud Computing provides offer scalable resources through virtualization for customers.
⑨ Virtual machines can be provisioned faster than physical hardware.
⑩ Virtualization divides physical servers into independent parts for better resource allocation.

Question 5- What is system call? Explain the system call flow with the help of diagram.

Ans:- A system call in an operating system (OS) is a mechanism that allows user level programs to request services of resources from the kernel, which is core part of OS. system calls provide an interface between applications and the underlying hardware and services of the computer system.

The system call can be described as follow with the help of below diagram-

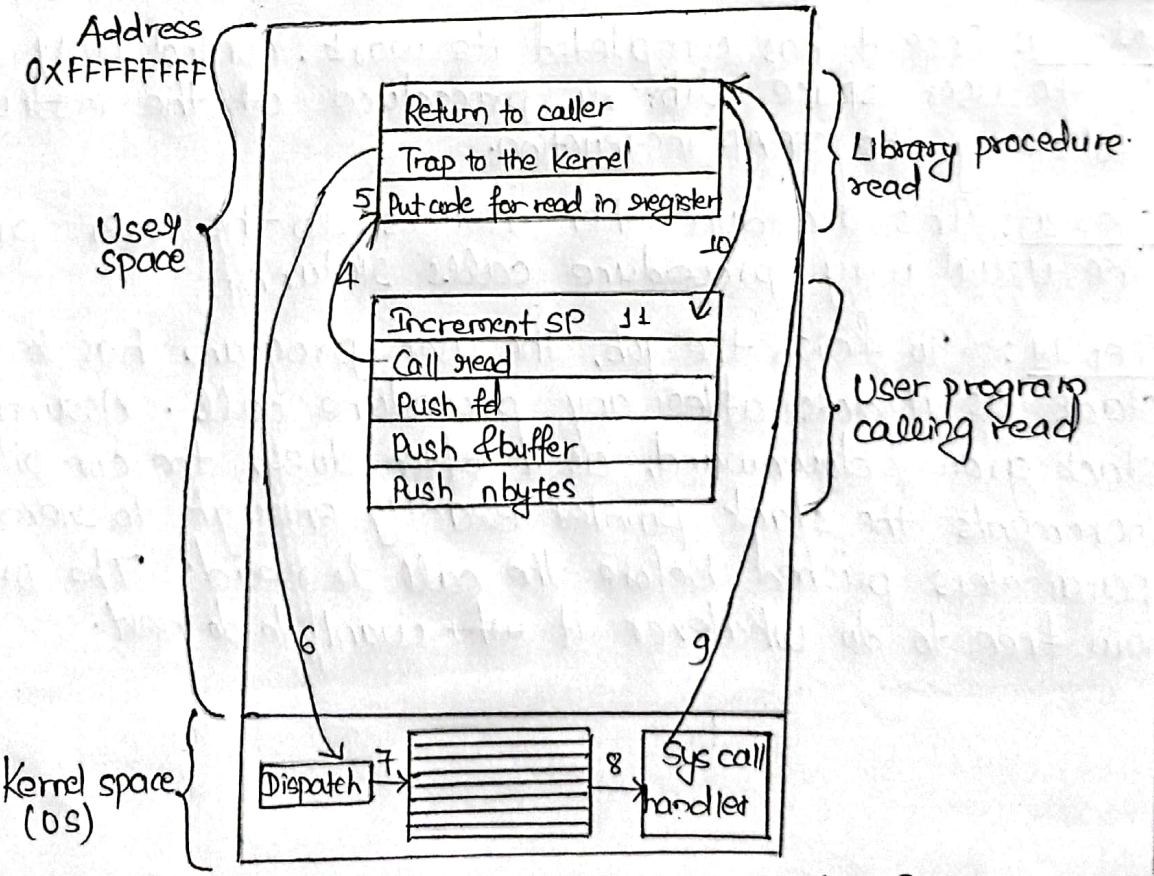


Fig: 11 steps for making system call read(fd, &buffer, nbytes)

System calls are performed in series of steps. To make this concept clear, Let us examine the read call.

Step 1-3: For calling the read library procedure, which actually makes the read system call, the calling program, first pushes the parameters onto the stack as shown in fig in reverse order.

Step 4: Then comes the Actual call to the library procedure. This instruction is the normal procedure-call instruction used to call all procedures.

Step 5: The library procedure, possibly written in assembly language, typically puts the system call number in a place where the OS expects it such as a Register.

Step 6: Then it executes a TRAP instruction to switch from user mode to kernel mode and start execution at a fixed address within the kernel.

Step 7: The kernel code that starts following the TRAP examines the system-call number and then dispatches to the correct system call handler usually via a table of pointers to system-call handlers indexed on system-call number.

Step 8: At that point the system-call handler runs.

Step 9: Once it has completed its work, control may be returned to the user-space library procedure at the instruction following the TRAP instruction.

Step 10: This procedure then returns to the user program in the usual way procedure calls return.

Step 11: To finish the job, the user program has to clean up the stack as it does after any procedure call. Assuming the stack grows downward, as it often does, the compiled code increments the stack pointer exactly enough to remove the parameters pushed before the call to read. The program is now free to do whatever it ~~who~~ want to do next.

Question 6- What do you mean by trap and interrupt? What is the use of each function?

=> Trap → A trap, also known as a software interrupt or exception is a synchronous event that is triggered by the execution of a specific instruction in a user-level program. It's a deliberate way for a program to request a service or action from the operating system.

Uses of trap →

① System calls: User-level programs use traps to request services from the operating system, such as reading from or writing to files, allocating memory or creating new processes.

② Error Handling: Traps can also be used for error handling. For eg. a-divide-by-zero operation triggers a trap, and the operating system can handle the error gracefully.

Interrupt → An interrupt is an asynchronous event that occurs external to the currently executing program. It is typically triggered by hardware devices (such as timer, keyboard, or disk controller) to request attention from the OS.

Uses of interrupt →

① I/O operations: Interrupts are used to signal that an I/O operation (like data transfer from a disk) has completed or that data is ready to be processed.

② Timer interrupts: A timer interrupts is used for preemptive multitasking, allowing the OS to regain control after a predefined time slice to ensure fair CPU scheduling.