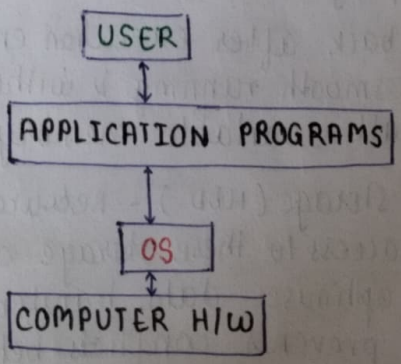
UNIT 1 - OVERVIEW OF OPERATING SYSTEM

OPERATING SYSTEM (OS) :-

- It is system s/w which manages, operates & communicates with the computer hardware & software
- It also provides common service for computer programs.
- Main Job - provide resource & services to user program.
- So without os, a computer would be useless 

- It acts as interface between the user & hardware of the computer
- It is also called as Resource Manager.



Need ?

① Identifying input from input device eg keyboard, mouse, etc.
② Sending o/p to o/p device eg monitor, printer, etc.
③ keeping track of files, Directories on the disk & Controlling peripheral devices
④ OS allocates memory to user program as per need.

OBJECTIVES OF OS

① convenience / User convenience
Provide user friendly interface, to make it easier for user to interact with the computer.
Following factors are considered:
- Good service
- End of use
- New programming language model
- Evolution
- User friendly OS

② Efficiency
Comp. sys. comprises of many resources All these are used efficiently only due to OS.

③ Ability to Evolve
An os should be constructed in such a way as to permit the effective development, testing & introducing of new systems functions without at the same time interfering with service.

④ Hardware Abstraction
User can fully utilise the computer hardware without having any accompaying difficulties.
It coordinates communication betn user program & computer hardware.

FUNCTIONS OF OS

- Processor, memory, file & devices are called resources.
- OS has different components each have its own defined i/p's & o/p

① Process Management
- provides control access to shared resources like file, memory, I/O, CPU.
- Controls execution of user appl^n
- creation, deletion & execution of user & system process.
- resume a process execution or cancel it.
- Scheduling a process, synchronization, interposes communication & deadlock handling for processes.

② Memory Management.
- Allocates primary & secondary mem.
- Keep track of how much memory is used by the process.

③ File Management
- Create & delete, files & directories
- Service to access files & allocates the storage space for files
- Keep back-ups & offer security of files.

④ Device Management
- Device drivers are opened, closed & written by OS.
- Communicate, control & Monitor device driver.

⑤ Protection and Security.
- Resources of system are protected by OS
- To provide protection, OS makes use of user authentication, file attributes (read, write, back-up data)

⑥ User Interface or Command Interpreter.
- OS acts as interface bet^n user & CH.

⑦ Booting the Computer
- Process of Starting or restarting the computer is known as booting
- Cold booting → PC switched off completely & then turned on
- Warm booting → using OS to restart PC

⑧ Performs Basic computer tasks
- Manages peripheral devices by os (mouse, keyboard, printers)

⑨ Resource Management
- refers to allocating, controlling & optimizing, utilization of various hardware & software resources of comp. sys.

•Key Resources that OS manage's
(i) CPU - efficient allocation of CPU time
(ii) RAM (memory) - provide mem to program when needed & takes it back after execution ensuring smooth running & without absorbing all available memory.
(iii) Storage (HDD) - ResourceM controls access to these storage devices, optimises data transfer & prevents conflicts bet^n programs
(iv) I/O Devices - RM ensures efficient utilization of I/O devices, ensuring data gets transferred correctly
(v) Network Resources - RM controls access to n/w bandwidth and shared printer & file servers

Role of OS as Resource Manager

EVOLUTION OF OS

Generation	Period	Development of os
Early Computers or First	1940's - 1950's	No OS Batch Processing sys.
Second	1950 - 1960	Multiprogramming Time sharing
Third	1960 - 1980	UNIX Real Time Systems MS-DOS GUI
fourth	1980's 2000's (present)	Windows, macos, Linux Mobile os, Cloud & Virtualisation

Types of OS.

① Batch os (factory assemby line)
- This type of os doesnot interact with computer directly.
- jobs with similar needs are batched together & executed sequentially.
- OS handles 1 batch at a time without user interacting during the process.
- Eg: Bank Statements, Payroll systems

② Time-sharing os (Restaurant with multiple tasks)
- Each task (table) is given some time to execute so that all tasks work smoothly.
- A time-sharing os allows multiple users to interact with comp. simultaneously by giving each user a small slice of the CPU time.

Analogy - multiple tables (users) are served in single kitchen (CPU), with the chef (os) ensuring that each table gets its turn.
- Eg: Unix, Multics.

③ Distributed os
- DOS manges a group of independent comp. & makes them appear as a single cohesive system.
- Diff company offices working together on a project, with each office handling its part but all contributing to the same goal.

Eg. Locus

④ Real Time os (RTOS)
- Processes the data & respond to events within strict time constraint.

Like an traffic control system where decisions must be made instantly to ensure safety, with no room for delays
Eg. robots, air traffic control sys, scientific experiments

⑤ Multiprogramming os
It is extension to batch os where CPU is always kept busy. Each process needs 2times - CPU time & I/O time
Eg. windows os, unix os

⑥ Multiprocessing os
Parallel computing is achieved. There are more than 1 processor present in sys. which execute more than 1 proces at same time.

⑦ Network os
NOS manages network resources, enabling computers to communicate, share files & manage device across a network.
Similar to post office network that manages flow of mail & packages both different locations.

DEVELOPMENTS LEADING TO MODERN OS

Categories of modern os
① Microkernel architecture
② Multithreading
③ Symmetric multiprocessing

④ Distributed OS
⑤ OO system Design

kernel is the "traffic controller" of computer, managing & coordinating all the "traffics" (appl^n), so everything runs smoothly!

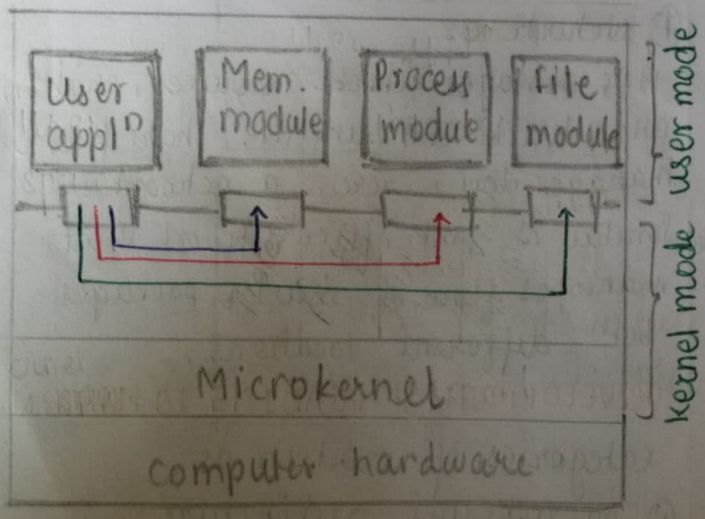
Difference betn Microkernel & Monolithic

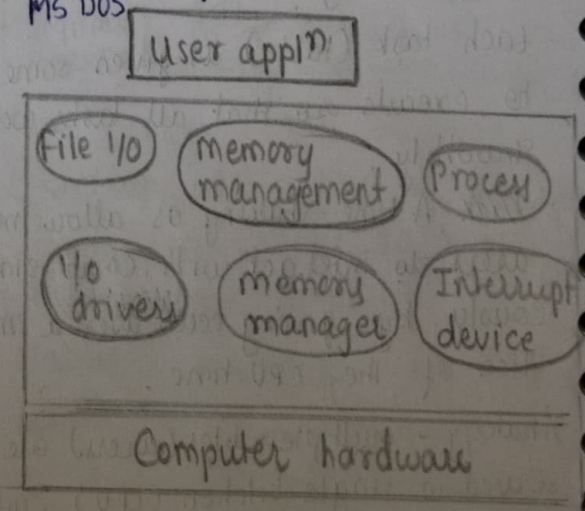
Microkernel :-

- User services & kernel services are kept in separate address space.
- OS is complex to design
- Smaller in size
- Easier to add new functionalities
- To design microkernel, more code is required.
- failure of one component does not effect the working of microkernel.
- Execution speed is low
- Easy to extend microkernel.
- Debugging is simple
- It is simple to maintain
- Message forwarding & context switching are required by microkernel
- Kernel only offers IPC & low level device management services.
- Eg. Mac OS

Monolithic kernel :-

- Both user services & kernel services are kept in the same add^n space
- OS is easy to design & implement
- larger in size
- Difficult to add new functionalities
- less code when compared to microkernel.
- failure of one component, leads to failure of entire system
- Execution speed is high
- Not easy to extend monolithic k
- Debugging is difficult
- Extra time & resources required for maintainance
- message passing & context switching are not required while kernel is working
- The kernel contains all os services
- Eg. Microsoft windows 95

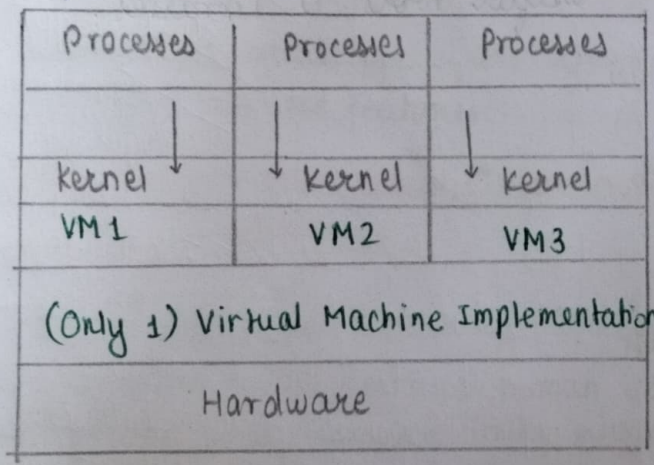




VIRTUAL MACHINES

- The fundamental idea behind VM is to abstract the hardware of a single computer (CPU, memory, disk drives, etc) into several different execution environments, thereby creating illusion that each separate execution environment is running its own private computer OR.

VM is a s/w program that acts like a separate computer system. It runs on top of physical computer h/w, creating an isolated environment with its own virtual CPU, memory, storage & OS.
VM's are digital comp. with physical computers.



Implementation:

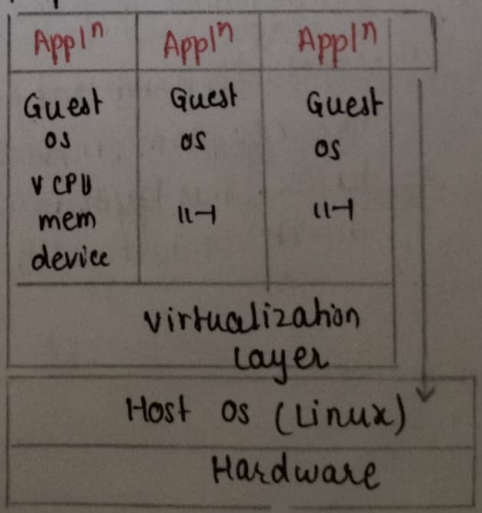
VM software - Runs in Kernel mode
VM itself (VM1, 2, 3) - Runs in User mode
Just as physical machine has 2 modes, however, so must VM
So we must have,
virtual user mode &
virtual kernel mode
Both of which run in a physical user mode

Benefits / Advantages of VM

① Multiple VM's run on single physic-al server, optimizing use of h/w resources (CPU, memory, storage)
② No additional physical h/w required to create VM's
③ VM's are isolated from each other, so issues in 1 VM not affect other VM's
④ Migration is easy betn physical servers.
⑤ Cloning VMs for testing or scaling purposes.
⑥ Reduce Hardware cost
⑦ Lower energy consumption due to reduced no. of physical servers.
⑧ Centralised control
⑨ Direct sharing of resource, possible

Examples:

① VMware
It is popular virtualization machine/s/w that allows users to create & run multiple VM on single physical comp. commonly used for testing & learning purposes.



② JVM (Java VM)
③ .NET Framework

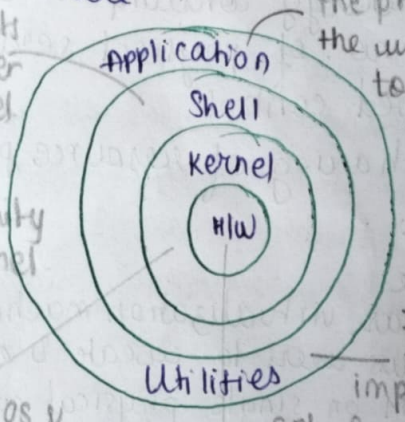
OS SERVICES

- Program Execution
- Input / Output Operation
- Error detection
- File & directory operation
- Communication
- GUI

INTRODUCTION TO LINUX OS

- developed in 1991 by Linus Torvalds
- It is a free operating system based on UNIX standards.
- It is multi-user, multi-tasking, time-sharing & monolithic kernel

Architecture



interacts with user & kernel

Hides all complexity of kernel

the programs that the user runs on top of architecture. This includes db appln, web browsers, etc.

Includes code that is used for implementing the spl. functions & appln. It Gives users variety of functionalities to manage os

Includes peripheral devices like CPU, RAM, input & o/p devices. This layer is in direct contact with H/W

core of OS & foundation of other s/w that works on top of it.

VMware -
lets you run many virtual computers on one real computer using a spl. software layer called hypervisor.

JVM -
lets you Java program run on any computer by translating Java bytecode into machine code.

.Net framework -
helps build & run Windows application by managing how code is executed & providing useful tools & libraries.