Parul® University
Vadodara, Gujarat

NAAC GRADE A++

Information and Communication Technology

Introduction to OS

Study Guide

Assistant Prof. Sumersing Patil
CSE-AI&DS, PIET
Parul University

1. Introduction

Concepts of Operating System

Software:-

- Program is a collection of code/instruction.
- Software is a collection of program.

Hardware:-

- Physical device is a collection of computer system which is called Hardware.

Example: **Processor, RAM, Hard disk, I/O devices.**

Types of Software:-

- Software is divide into 3 types:
    1. System software
    2. Utility software
    3. Application software
- **System Software**
- The software which is used to perform all types of system level tasks of computer is called system software.

For example:

- Compiler
- Operating system
- Interpreter
- Linker
- Loader

- **Utility Software**

- The software, which provide an additional meaning to the computer system.

For Example:-

  - Calculator
  - MS-paint
  - Browser
  - Notepad
  - Media Player

- **Application Software**
- The software which is created by users, using the different high level language and database system for any special purpose.

For Example:-

  - Library Management system
  - Banking Software
  - Ticket Reservation system

- **What is an Operating System?**
- An operating system (OS) is a collection of system software that manages computer hardware resources and provides common services for computer programs.
- A program that acts as an intermediary/interface between a user of a computer and the computer hardware.
- **Goals of an Operating System**
- Simplify the execution of user programs and make solving user problems easier
- Use computer hardware efficiently
- Allow sharing of hardware and software resources.
- Make application software portable and flexible
- Provide isolation, security and protection among user programs
- Improve overall system reliability like Error confinement, Fault tolerance, Reconfiguration.
- **Generations of Operating Systems**
  - It's also known as history of Operating systems.
  - Which can be divided in 4 generations
- **Types of Operating Systems**
  - Simple Batch System

- Multiprogramming Batch System
- Multitasking system
- Multiprocessor System
- Distributed Operating System

Real-time Operating System

SIMPLE BATCH SYSTEMS

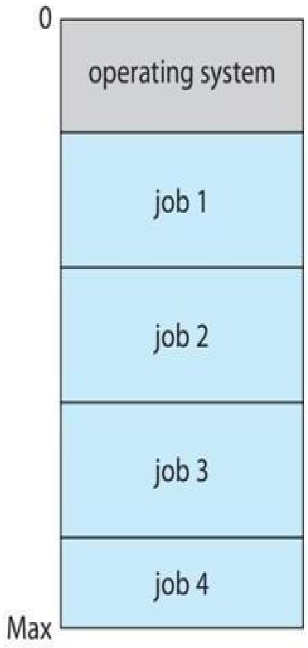
- **No direct interaction** between user and computer.
- The user has to **submit a job** (written on cards or tape) to a computer operator.
- Then computer operator places a **batch of several jobs** on an input device.

Advantages

- Increased performance - next job start as the previous job finished.
- Suitable for executing large jobs that need little interaction

Disadvantages

- Zero interaction between user and computer.
- No mechanism to prioritize processes.
- **Multiprogramming Batch Operating System**



- Several jobs are kept in main memory at the same time, and the CPU is multiplexed among them.
- Multiprogramming increases CPU utilization

- Multiple jobs are loaded into main memory and one is selected from pool for execution by CPU
- If at some point program in progress requires service of a peripheral device, the control of CPU is given to next job which is in main memory
- So, CPU is always executing some program instead of waiting.

Advantages

- High CPU utilization, so CPU never sits idle, if there are jobs available
- Many programs are allotted CPU almost simultaneously.
- Provides better resource utilization (Memory, I/O, CPU)
- More then one process can be executed simultaneously by user.

Disadvantages

- CPU scheduling is required.
- Memory management is required, to accommodate many jobs in memory
- Multiprogramming does not support interaction with users
- **Multitasking/Time Sharing System(TSS)**
- Multiprogramming does not support interaction with users, TSS extends multiprogramming to handle multiple interactive jobs
- TSS uses CPU scheduling & multiprogramming to provide economical interactive systems of two or more users.
- Each user is given a time-slice for executing his job in Round-Robin Fashion (Every process will be given equal amount of CPU one by one in sequence). Job continues until the time slice ends.
- The CPU is multiplexed among several jobs that are kept in main memory.
- TSS allows more frequent context switches from one user to the next (when time-slice of particular process ends it switches to the next for given time slice duration)
- This gives each user the impression that the entire computer is dedicated to his use only, whereas actually one computer is being shared among many users.

Advantages

- Provides Quick Response
- Reduces CPU idle time

Disadvantages

- Security & Integrity of user’s program & data is needed.

- If lots of users & applications are running then it may hang up the system. So, high configuration of hardware is required.
- **Multiprocessor/Parallel System**
- **Multiprocessor systems** with more than one CPU works in close communication.
- **Tightly coupled system** – processors share memory and I/O devices, bus, system and communication usually takes place through the shared memory.
- A multiprocessor system comprises of **several processors** that share a **common physical memory**.
- Multiprocessor system delivers **higher computing power and speed.**
- In multiprocessor system all processors function under **single operating system.**

Advantages

- **Increased throughput:** No. of jobs executed per unit time increased as there are more no. of processors.
- **Economical:** Buying one system with 3 CPU is cheaper than 3 systems with 3 different CPUs. The processors can share peripherals, cabinets and power supplies.
- **Increased reliability:** The failure of one processor will not stop the system, it functions with other available processors.
- **Real Time System**
- A real-time operating system (RTOS) promises a certain capability within a specified time constraint.
- It is defined as an operating system known to give maximum time for each of the critical operations that it performs, like OS calls and interrupt handling.

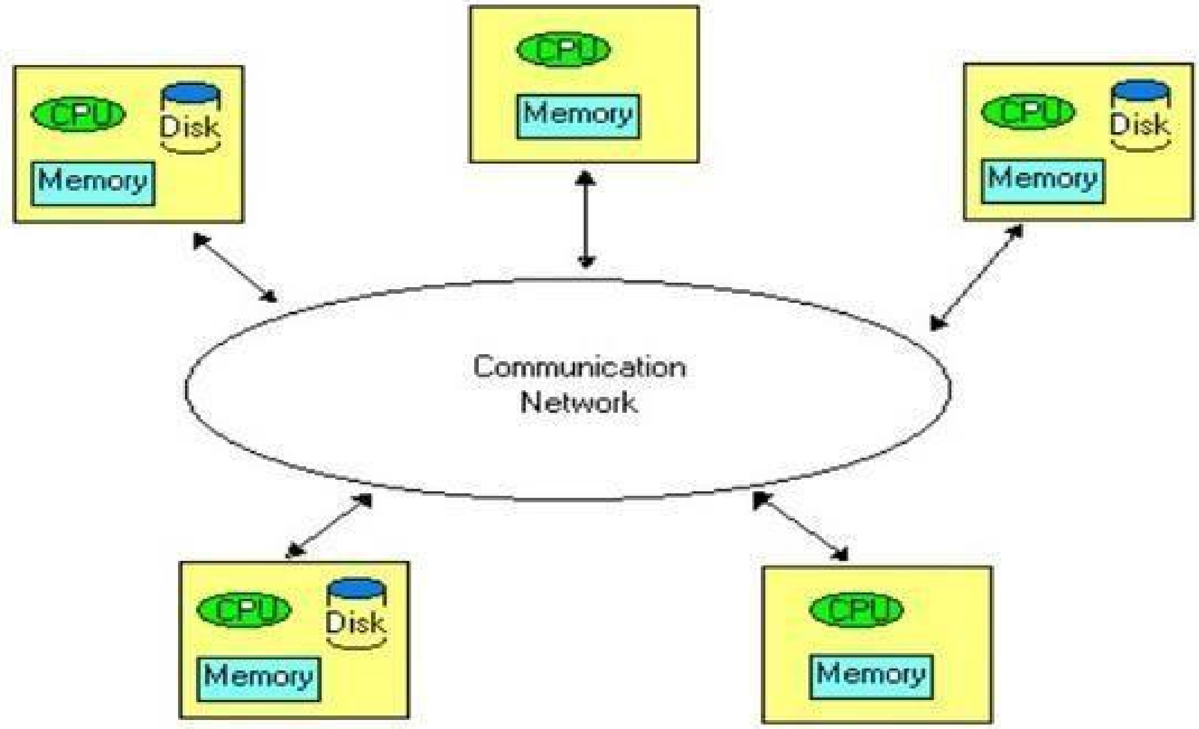
Hard real-time system

- The Real-Time Operating system which guarantees the **maximum time for critical operations** and complete them **on time** are referred to as Hard Real-Time Operating Systems.
- If the system fails to meet the deadline even once the system is considered to have Failed.
- E.g. Defence applications, nuclear system etc. Missing deadlines creates hazards.

Soft real-time system

- The critical task will get priority over other tasks, but **no assurity of completing it in a defined time**. These systems are referred to as Soft Real-Time Operating Systems.
- It is less restrictive type of OS. even if the system fails to meet the deadline, the system is not considered to have failed. In this case the results of the requests are not worthless.

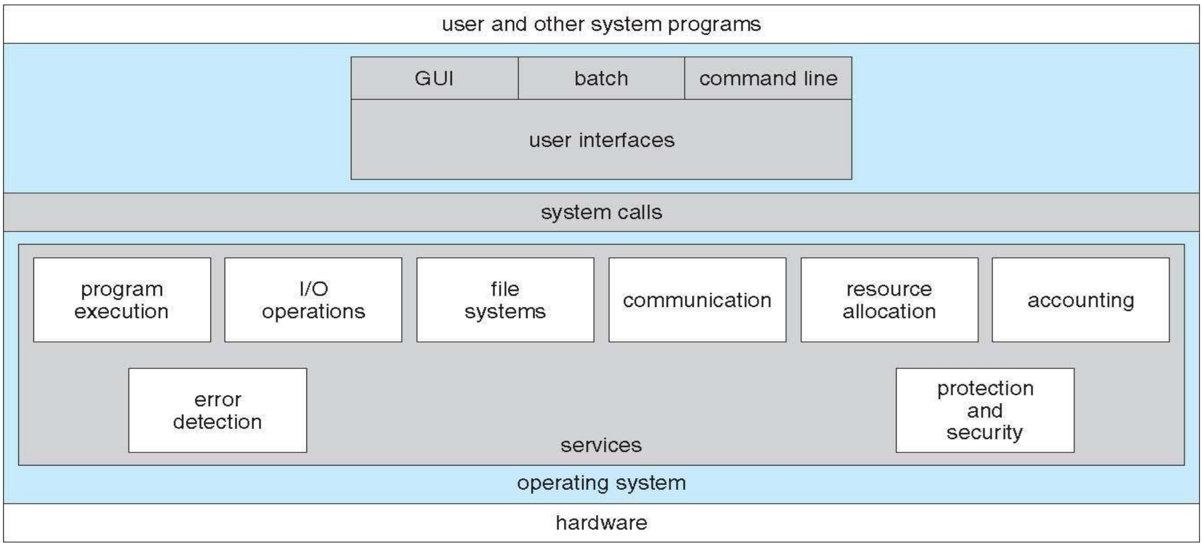
- E.g. Audio-Video streaming etc.
- **Distributed System**
- Distribute the computation among several physical processors.
- Distributed OS is an OS that runs on several machines and it controls the resources of several machines.
- Loosely coupled system – each processor has its own local memory; processors communicate with one another through various communications lines, such as high-speed buses or telephone lines.



Advantages

- Resources Sharing
- Computation speed up due to load sharing . So, Short response time and higher throughput.
- Higher Reliability: Degree of tolerance against failure
- Incremental Growth : to extend functionality of a system by simply adding additional resources to the system

Operating System Services



- ***User Interface:*** Almost all operating systems have a user interface (UI). Varies between Command-Line (CLI), Graphics User Interface (GUI), Batch Interfaces

- ***Program execution:*** The system must be able to load a program into memory and to run that program, must be able to end execution, either normally or abnormally (indicating error)

- ***I/O operations:*** A running program may require I/O, which may involve a file or an I/O device, user programs cannot execute I/O operations directly, the operating system must provide some means to do I/O.

- ***File-system manipulation:*** Programs need to read and write files and directories, create and delete them, search them, list file Information, permission management; allow or deny access to files/directories based on file ownership.

- ***Communications:*** Exchange of information between processes executing either on the same computer or on different systems tied together by a network. Implemented via shared memory or message passing.

- ***Error detection*** – OS needs to be constantly aware of possible errors

- May occur in the CPU and memory hardware, in I/O devices, in user program

- For each type of error, OS should take the appropriate action to ensure correct and consistent computing

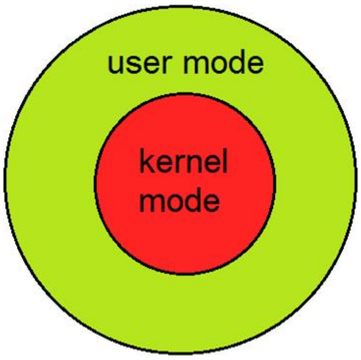
- Debugging facilities can greatly enhance the user's and programmer's abilities to efficiently use the system

- ***Resource allocation:*** When multiple users or multiple jobs running concurrently, resources must be allocated to each of them. Many types of resources - Some (such as CPU cycles, main memory, and file storage) may have special allocation code, others (such as I/O devices) may have general request and release code.

- ***Accounting:*** To keep track of which users use how much and what kinds of computer resources. Used for accounting or usage statistics.
- ***Protection*** involves ensuring that all access to system resources is controlled
- ***Security*** of the system from outsiders requires user authentication (by password), extends to defending external I/O devices (eg. Modems, network adapter from invalid access attempts .
- ***Protection and security:*** The owners of information stored in a multiuser or networked computer system may want to control use of that information, concurrent processes should not interfere with each other.
- **System Calls**
- System call is a request made by user program in order to get the service of an operating system.
- When a program in user mode requires access to RAM or a hardware resource, it must ask the kernel to provide access to that resource. This is done via something called a system call.

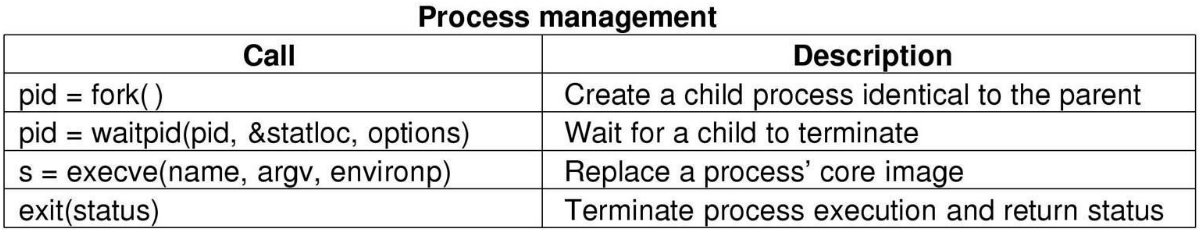
Kernel Mode

- When CPU is in kernel mode, the code being executed can access any memory address and any hardware resource.
- Hence kernel mode is a very privileged and powerful mode.
- If a program crashes in kernel mode, the entire system will be halted.



User Mode

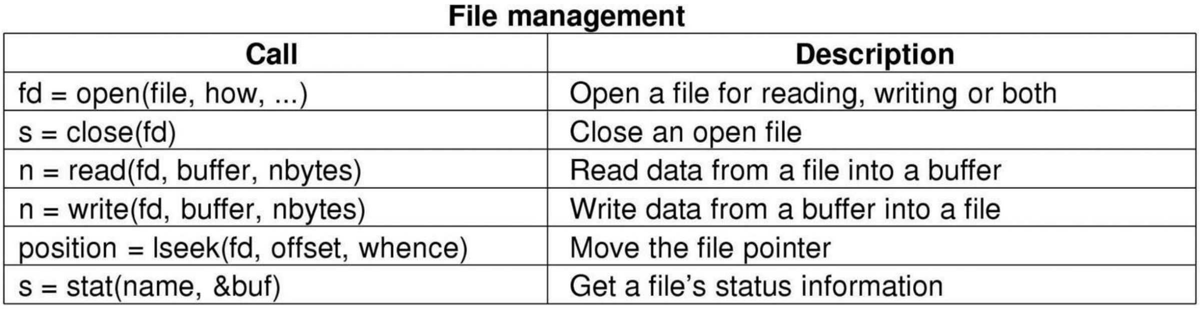
- When CPU is in user mode, the programs don't have direct access to memory and hardware resources.
- In user mode, if any program crashes, only that particular program is halted. That means the system will be in a safe state even if a program in user mode crashes.
- Hence, most programs in an OS run in user mode.
- **System Calls For Process Management**



Process management

Call	Description
pid = fork( )	Create a child process identical to the parent
pid = waitpid(pid, &statloc, options)	Wait for a child to terminate
s = execve(name, argv, environp)	Replace a process' core image
exit(status)	Terminate process execution and return status

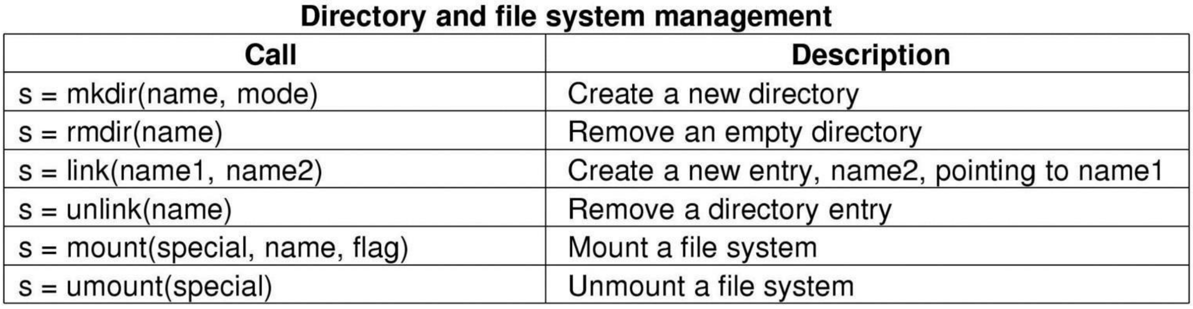
System Calls For File Management



File management

Call	Description
fd = open(file, how, ...)	Open a file for reading, writing or both
s = close(fd)	Close an open file
n = read(fd, buffer, nbytes)	Read data from a file into a buffer
n = write(fd, buffer, nbytes)	Write data from a buffer into a file
position = lseek(fd, offset, whence)	Move the file pointer
s = stat(name, &buf)	Get a file's status information

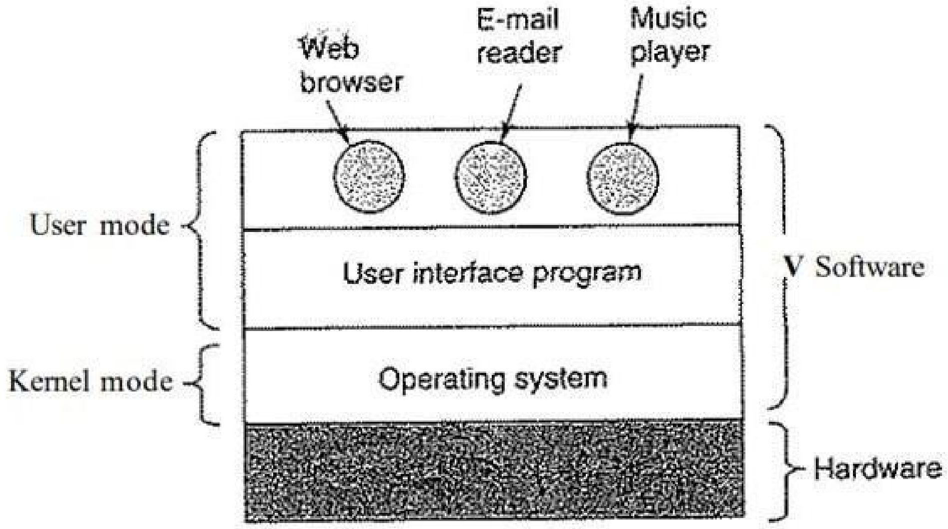
System Calls For Directory Management



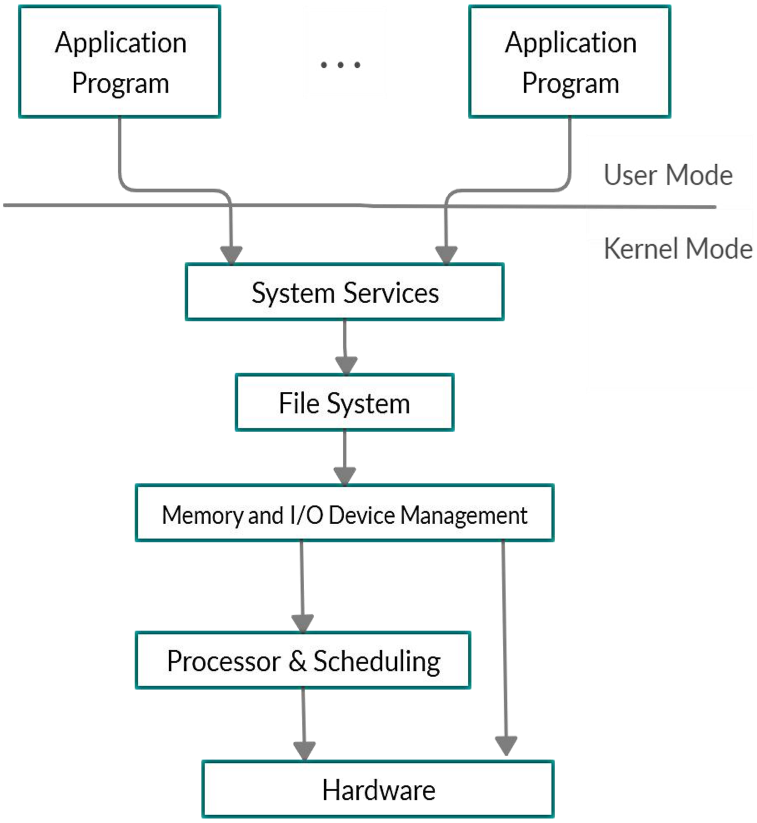
Directory and file system management

Call	Description
s = mkdir(name, mode)	Create a new directory
s = rmdir(name)	Remove an empty directory
s = link(name1, name2)	Create a new entry, name2, pointing to name1
s = unlink(name)	Remove a directory entry
s = mount(special, name, flag)	Mount a file system
s = umount(special)	Unmount a file system

Operating System layered structure

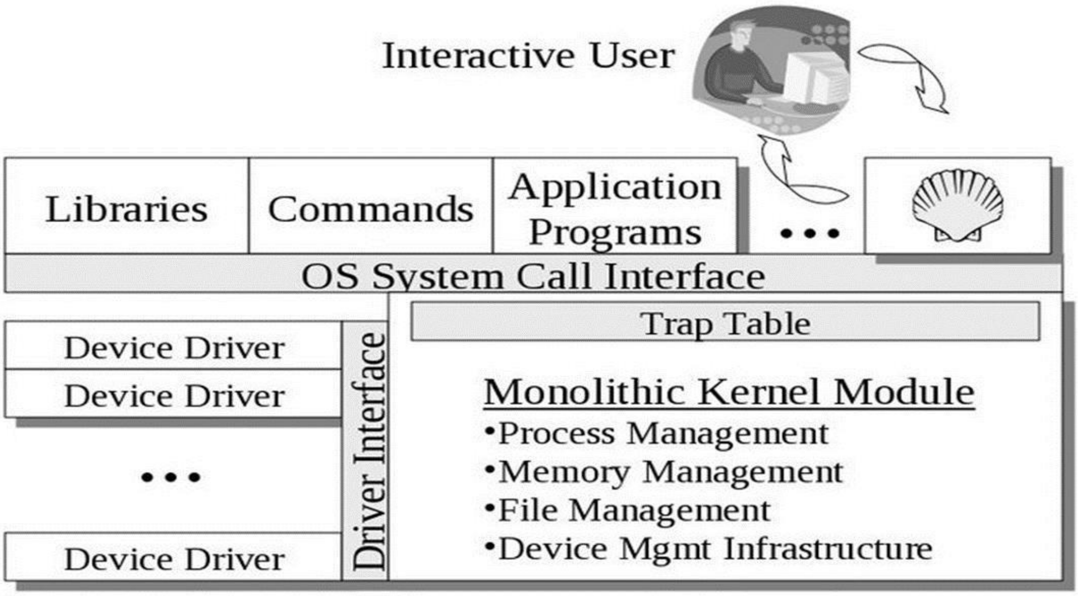


- With the layered approach, the bottom layer is the hardware, while the highest layer is the user interface.
- Advantage is simplicity of construction and debugging.
- The main difficulty is defining the various layers.
- The main disadvantage is that the OS tends to be less efficient than other implementations



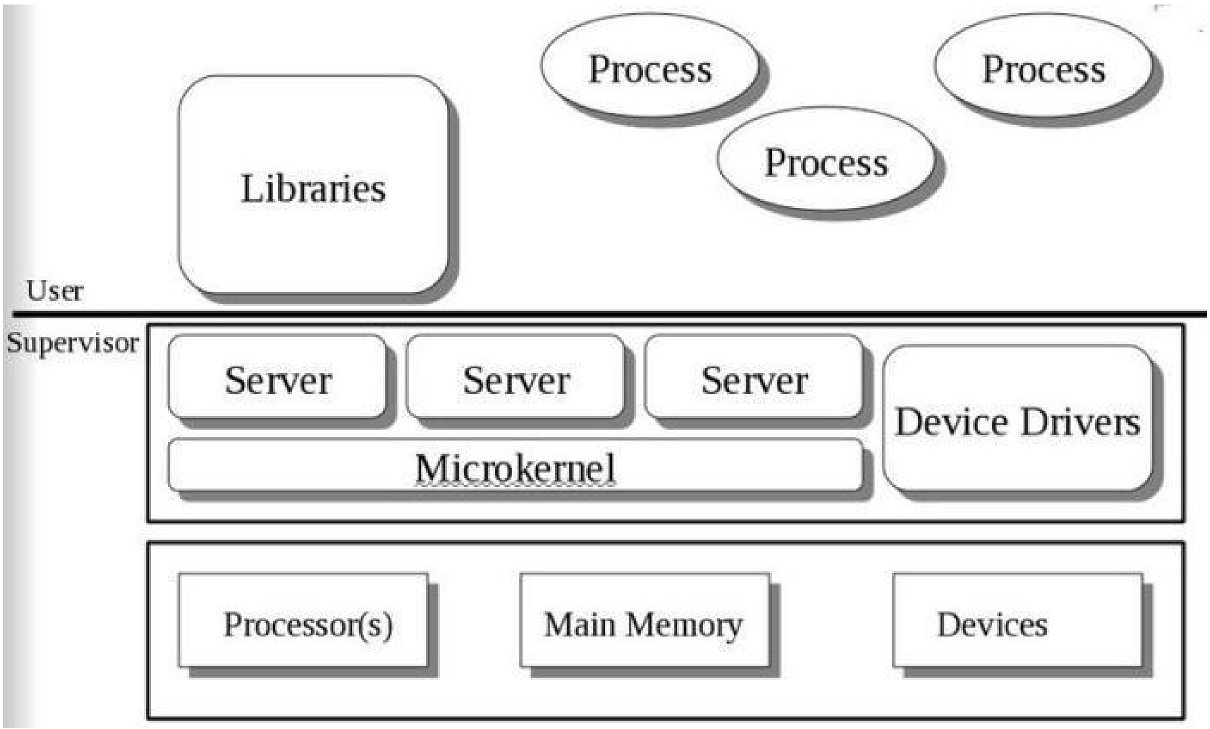
Monolithic Approach

- Functionality of the OS is activated with simple function calls within the kernel, Monolithic kernel is one large program.
- Device drivers are loaded into the running kernel and become part of the kernel.



Microkernel Approach

- Microkernel structures the OS by removing all unnecessary parts of the kernel and implement them as system and user level programs.
- They offers minimal process and memory management, and a communications facility.
- Communication between components is done by message passing.



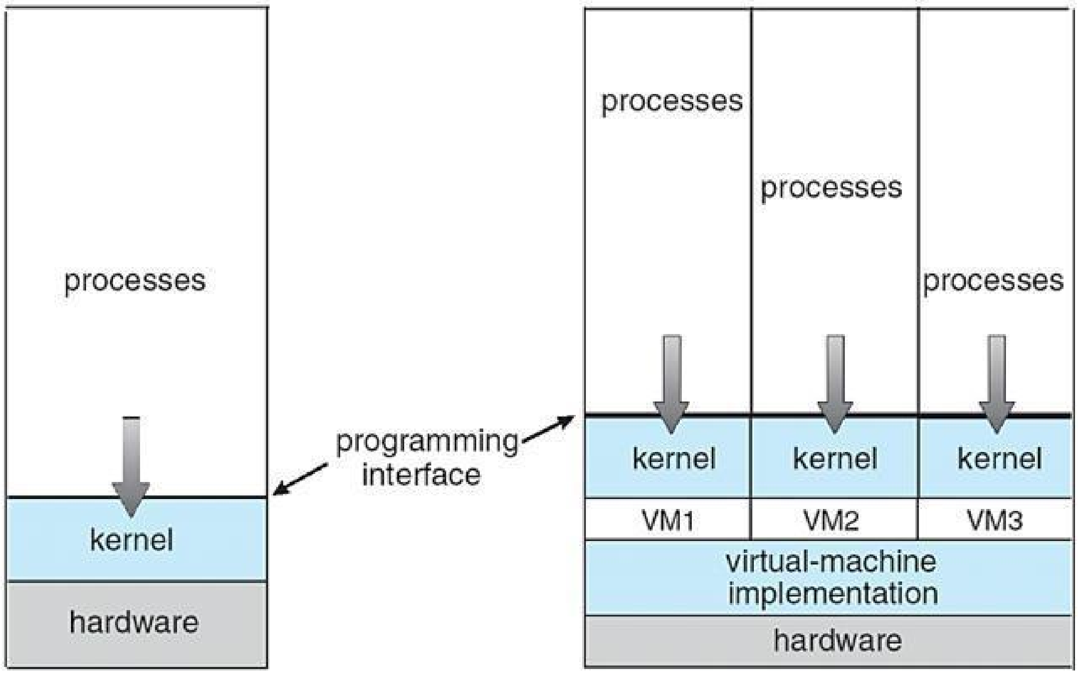
Advantage

- Operating system can be easily extended

- Kernel is smaller, so very few changes are required in it.
- It offers more security and reliability.

Disadvantage

- It has poor performance due to increased system overhead of message passing.
- **Virtual Machine**
- Virtual machine does abstract the hardware of a single computer (the CPU, Memory, Disk drives, Network Interface Cards) **into several different execution** environments and thereby **creating the illusion** that each separate execution environment is running its own PC/environment.



- Virtual Machine(VM) is also known as a guest machine, which is created within another computing environment known as a “host”.
- Multiple VM can be present within a single host at one time.