

OPERATING SYSTEM OVERVIEW

- * What is an Operating System?

An OS is a program that controls the execution of application programs and acts as an interface between application and the computer hardware. It can be thought of as having three objectives.

- Convenience : An OS makes a computer more convenient to use.
- Efficiency : An OS allows the computer system resources to be used in an efficient manner.
- Ability to Evolve : An OS should be constructed in such a way as to permit the effective development, testing, and introduction of new functions without interfering with existing ones.

* OPERATING SYSTEM SERVICES

An OS provides services to both the users and to the programs

- It provides program an environment to execute
- It provides user the services to execute the program in a convenient manner.

Following are few common services provided by an OS

• Program Execution

OS handles many kind of activities from user programs to system programs like printer spooler, name servers, file server, etc each of these activities is encapsulated as a process.

- Loads program into memory
- Executes the program
- Handles program's execution
- Provides a mechanism for process synchronization
 - " " " " " communication
 - " " " " " deadlock handling

• I/O Operation

An OS manages the communication between user and device drivers

- I/O operation means read/write operation with any file or any specific I/O device
- OS provides the access to the required I/O device when required

• File System Manipulation

A file system is normally organized into directories for easy navigation and usage. These directories may contain files and other files. Following are the major activities of an OS with respect to the file management

- Program needs to read/write a file
- The OS gives the permission to the program for operation on file
- Permission varies from read-only, read-write, denied and so on
- OS provides an interface to the user to create/delete files and directories
- & also create backup of file

System

o Communication

The OS handles routing and connection strategies and the problem of contention and security. Following are the major activities of OS with respect to communication -

- Two processes often require data to be transferred between them
- Both the processes can be on one computer or on diff computer, but are connected through computer network
- Comm? may be implemented by two methods; either by Shared Memory or by Message Passing.

o Error Handling

Errors can occur anywhere and anytime. An error may occur in CPU, in I/O devices or in the memory/hardware. Following are the major activities of an OS w.r.t error handling -

- The OS constantly checks for possible errors
- The OS takes an appropriate action to ensure correct & consistent computing.

• Resource Management

In case of a multi-user or multi-tasking environment, resources such as an main memory, CPU cycles and files storage are to be allocated to each user or job. Following are the major activities of an OS w.r.t resource management.

- The OS manages all kinds of resources using schedulers
- CPU scheduling algorithms are used for better utilization of a CPU

◦ Protection

Protection refers to a mechanism or a way to control the access of programs, processes, or users to the resources defined by a computer system. Following are the major activities of an OS w.r.t protection.

- The OS ensures that all access to system resources is controlled
- The OS ensures that external I/O devices are protected from invalid access attempts.
- The OS provides authentication features for each user by means of passwords.

• KERNEL

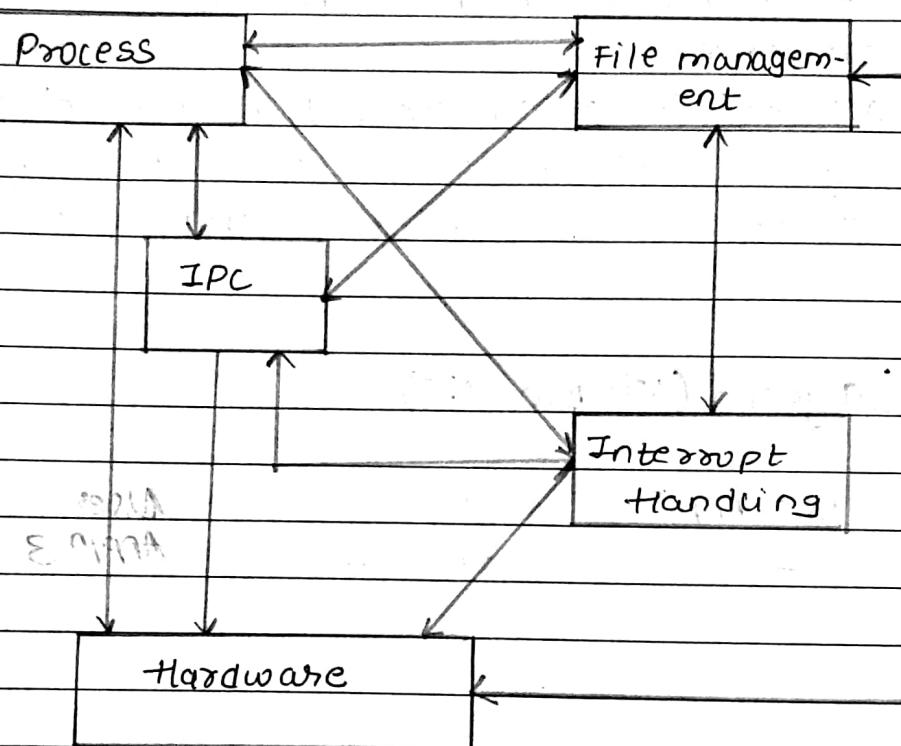
Kernel is the core part of the Linux. It is responsible for all major activities of this OS. It consists of various modules and it interacts directly with the underlying hardware. Kernel provides the required abstraction to hide low level hardware details to system or application programs.

(i) MONOLITHIC KERNEL

A Monolithic kernel is an OS software framework that holds all privileges to access I/O devices, memory, hardware interrupts and the CPU stack. Monolithic kernels tend to be larger than other kernels because they deal with so many aspects of computer processing at the lowest level, and therefore have to incorporate code that interfaces with many devices, I/O and interrupt channels and other hardware operators.

Structure

User APPIN 1	User APPIN 2	User APPIN 3
-----------------	-----------------	-----------------



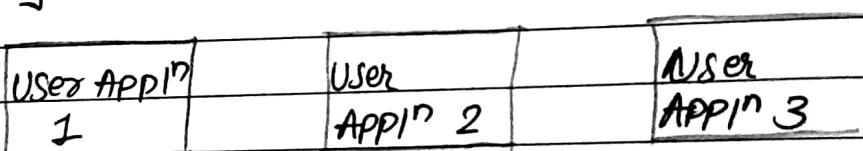
Advantages

- 1] Code is accessible to all components of OS & can be executed w/o much restrictions making the overall system fast.
- 2] Direct communication between components which increases the efficiency

Disadvantages

- 1.] Monolithic kernel are more prone to errors as user process runs in the same address space as that of a kernel.
- 2.] Maintaining the code is difficult.
- 3.] It is harder to port because of dependencies on code.
- 4.] Adding / Removing any feature in monolithic system are quite difficult and often require rewriting and recompiling whole code.

* Layered (Windows NT)



Layer N

Level
of

Abstrac-
tion

Layer 0

Hardware

This is an important architecture of operating system which is meant to overcome the disadvantages of early monolithic systems. In this approach, OS is split into various layers such as that all the layers perform different functionalities.

Each layer can interact with the one just above it and the one just below it. Lowermost layer which directly deals with the base hardware is mainly meant to perform the functionality of I/O communication.

Example: Microsoft NT

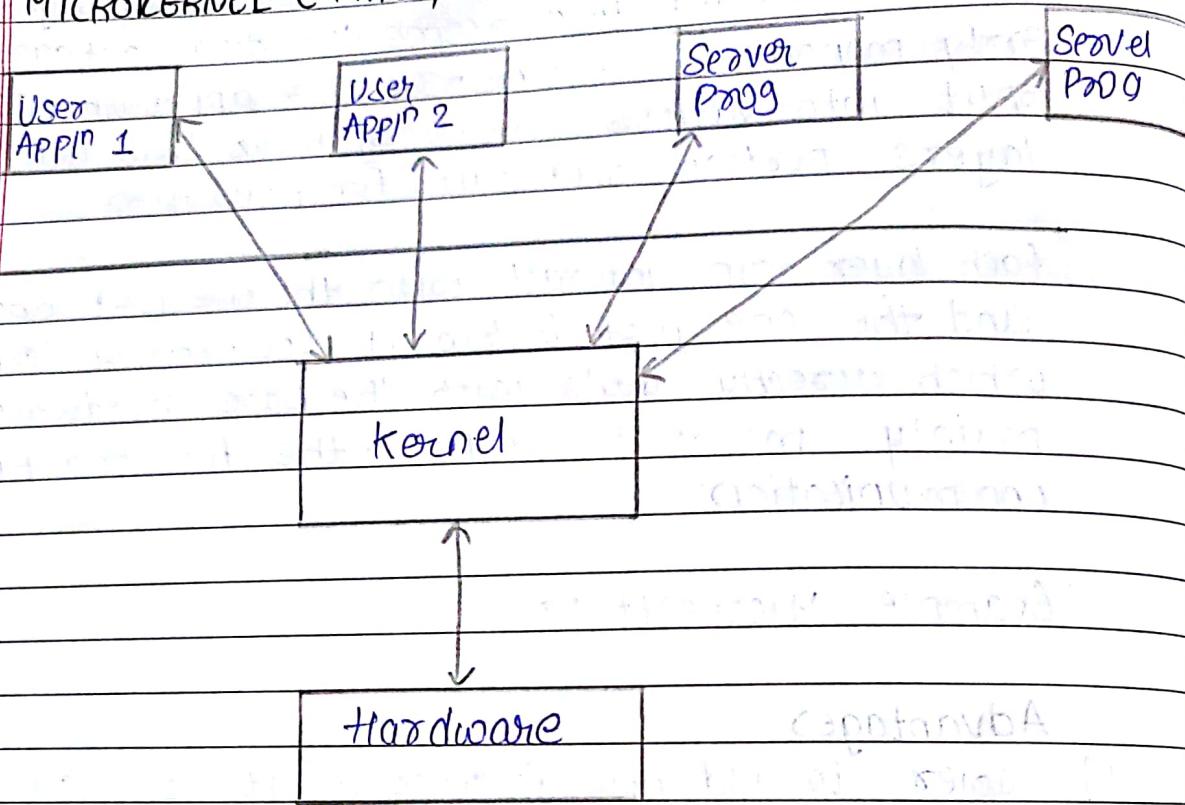
Advantages

- 1.] Easier to add new features as it will change only 1 layer w/o affecting others
- 2.] Any bug is restricted to the level itself
- 3.] Easy to add new layers

Disadvantages

- 1.] More layers lead to more overhead
- 2.] If functionality is not properly divided among the layers it may overburden the layer & overall decreases the performance

MICROKERNEL (MAC, MINIX, OS X)



The basic ideology in this architecture is to keep the kernel as small as possible. We know that kernel is the core part of the OS and hence it should be meant for handling the most important services only.

In this architecture, only the most important services are put inside the kernel and rest of the OS service are present in the system application programs. Microkernel is solely responsible for the 3 most important services of operating system namely:

- 1) Inter-Process Communication
- 2) Memory Management
- 3) CPU Scheduling

Advantages

- Easier to add new functionalities or to modify existing ones.
- System is reliable as fault in one server process do not halt the working of entire system.
- It is easily portable.
- Minimum functionality within the kernel is small and easy to maintain.

Disadvantages

- Execution Rate is Slow.
- * SHELL

A shell is software that provides an interface for an operating system's user to provide access to the kernel's services.

On Unix-based or Linux-base OS, a shell can be invoked through the shell command in the CLI, allowing users to direct operations through computer commands, text or script.

Shells also exist for programming languages, providing them with autonomy from the OS and allowing cross-platform compatibility.

PROCESS

A process is a program in execution. Process is not as same as program code but a lot more than it. A process is an 'active' entity as opposed to program which is considered to be a 'passive' entity. Attributes held by process include hardware state, memory, CPU etc.

Process Memory is divided into 4 sections for efficient working:

- 1.] The Text Section is made up of the compiled program code, read in from non-volatile storage when the program is launched
- 2.] The Data Section is made up of the global and static variables, allocated and initialized prior to executing the main
- 3.] The Heap is used for the dynamic memory allocation, and is managed via calls to new, delete, malloc, free, etc.
- 4.] The Stack is used for local variables. Space on the stack is reserved for local variables when they are declared

OPERATING SYSTEM EVOLUTION

1.] Serial Processing

The Serial Processing OS are those which performs all the Instructions into a Sequence Manner or the Instruction those are given by the user will be executed by using the FIFO manner. All the Instruction those are entered first in the system will be Executed first and the Instruction those are Entered Later will be Executed Later. For running the Instructions the Program Counter is used which is used for Executing all the Instructions.

2.) Batch Operating System

The users of BOS do not interact with the computer directly. Each user prepares his job on an off-line device like punch cards and submits it to the computer operator. To speed up processing, jobs with similar needs are batched together and run as a group. The programmers live their programs with the operator and the operator then sorts the similar requirements into batches.

Problems:

- Lack of interaction between user and job
- Difficult to provide desired priority
- CPU is often idle, because the speed of the mechanical I/O devices is slower than CPU

3) Time-Sharing OS

Time-Sharing is a technique which enables many people, located at various terminals, to use a particular computer system at the same time. Time-Sharing or multi tasking is a logical extension of multiprogramming. Processors time, which is shared among multiple user simultaneously, is termed as time-sharing.

Advantages

- Provides the adv of quick response time
- Avoids duplication of software
- Reduce CPU idle time

Disadvantages

- Problem of reliability: The multiprogramming
- Problem of data communication
- Question of security and integrity of user programs and data

4.0] MULTIPROGRAMMING OS

To overcome the problem of underutilization of CPU and main memory, the multiprogramming was introduced.

In multiprogramming system, when one program is waiting for I/O transfer; there is another program ready to utilize the CPU. So it is possible for several jobs to share time of the CPU.

But it is important to note that multiprogramming is not defined to be the execution of jobs at the same instance of time. Rather it does mean that there are a number of jobs available to the CPU and a portion of one is executed then a segment of another and so on.