LIC
भारतीय जीवन बीमा निगम
LIFE INSURANCE CORPORATION OF INDIA

OS-ESD

Chapter : 1

Introduction to Operating System and System Structures.

1] What is operating system? What are the main goals of operating system?

→ An Operating system is a software or program that acts as an interface between the user and the hardware.

Main Goals are :-

1) User Interface
2) Program execution
3) I/O operations
4) File system manipulation
5) Communication
6) Error detection
7) Resource allocation
8) Accounting
9) Protection and security.

1) User Interface :-
→ Almost all operating systems have user interface.
→ It varies from command line (CLU), graphic user interface (GUI), batch.

2) Program execution :- The system must be able to load a program, run it, and the execution either normally or abnormally.

3) I/O opirations :- A running program requires I/O operations which may include a file or an I/O device.

4) File system manipulation :- File system is of particular interest.
Programs need to read and write files and directories, create and delete files, share files, make list of them and permission management.

5) Communication :- Processes may exchange information, on the same computer or different computers.
Communications maybe via shared

memory or through message passage.

6) Error detection :- OS needs to be constantly aware of the errors that may occur in the system.

→ It may occur in the CPU or I/O device or in user program.

→ For each type of error, it must be able to take appropriate action to ensure correct and consistent computing.

→ Debugging facilities can greatly enhance user's and programmer's ability to efficiently use the system.

7) Resource allocation :- When multiple users or jobs are running concurrently, the resources must be allocated to them. The resources may vary, like CPU cycles file storage, main memory.

8) Accounting :- To keep track of which user uses ~~what~~ ~~or~~ how much & what kinds of computer resources.

1) Protection and Security :-

Protection :- It involves all access to the system resources is controlled.

Security :- Security of system from outsiders requires user authentication and extends upto defending external I/O devices from invalid access attempts.

2] What are system calls? Explain with appropriate example.

→ Programming ~~services~~ interface to the services provided by the OS are called system calls.

→ They are usually written in high level language like C or C++.

→ They are mostly accessed by programs via high-level Application programming Interface (API) than direct system call use.

→ The most three common API's are

Win32 API for Windows, POSIX API for POSIX based systems (including ~~al~~ virtually all versions of Linux, ~~Unix~~ UNIX, MAC OS X), JAVA API for Java

Virtual machine (JVM)

→ Example for system calls.

Example to copy ~~or~~ the contents of one file to the other.

source file → Destination file

Example system call sequence.

Acquire the input file name
Write prompt on the screen
Accept input.
Acquire output file name
Write prompt on the screen
Accept input.
Open the input file
If it doesn't exist, abort.
Create output file.
If it exists, abort
Loop
Read from input file
Write into output file.
~~Read until~~ Until read fails

Close the output file.
Write completion message to the screen
Terminate normally.

* Implementation :

→ Whenever a program in user mode requires ~~requiring~~ access to the RAM or hardware resources, it must ask kernel to give the access those recources. This is done by system calls.

→ When a program makes a system call, it is switched from user mode to kernel mode and this is called context switching.

→ Kernel provides access to the resources that are requested.

→ Another system call is made which changes mode from kernel mode to user mode.

→ System calls are made in following situations :-

1) Opening, closing, creating & deleting files in a system.
2) Managing new process.
3) Requesting access to hardware device.

* Types of System calls:-

1) Process Control
2) File management
3) Device management
4) Information maintainence
5) Communication
6) Protection.

1) Process control
→ Create & delete process
→ end, abort
→ get process attributes
→ set process attributes.

2) File Management :-
→ Create file, delete file
→ Open file, close file
→ Read, write file
→ get file attributes
→ set file attributes.

3) Device management :-
→ Request device, release device
→ Read, write, reposition
→ get device attributes
→ set device attributes

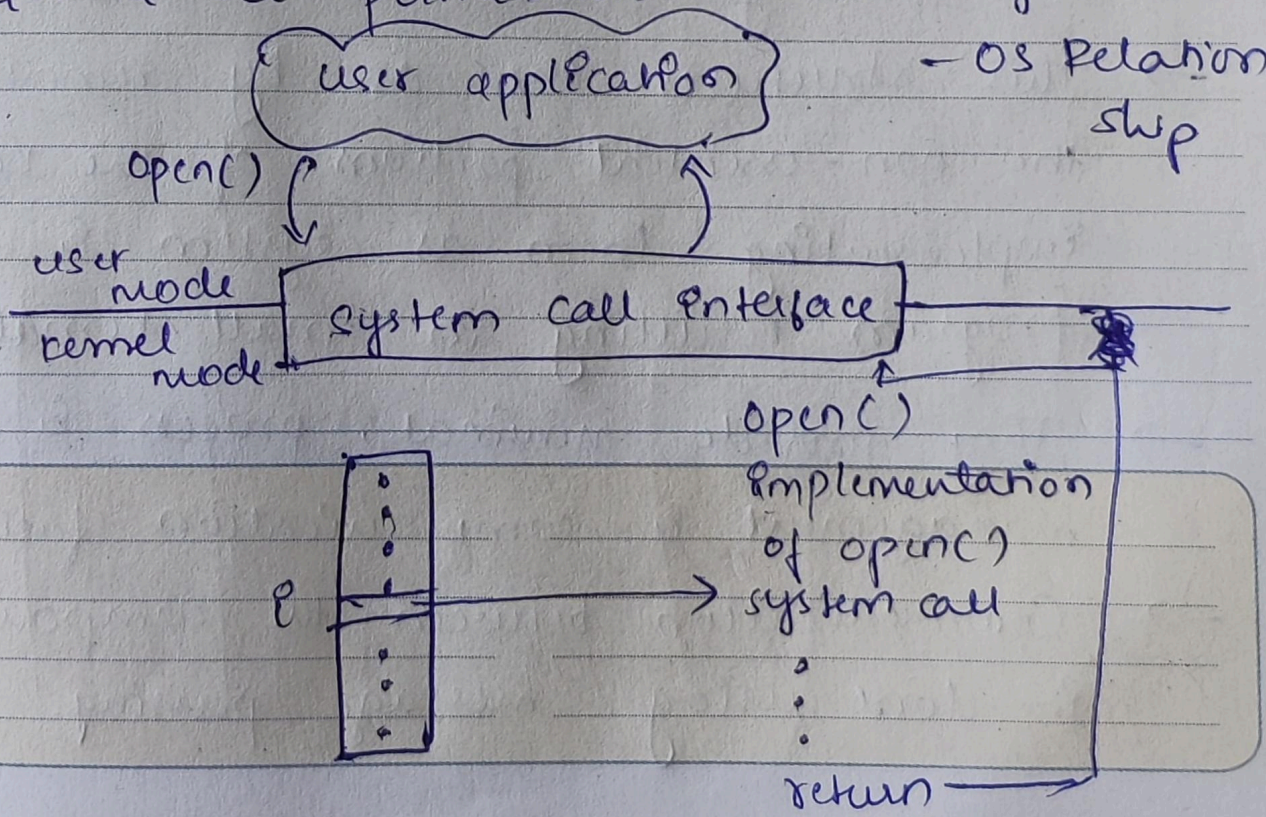
→ Logically attach or detach devices

4) Information maintainence :-
→ get time & date
→ set time & date
→ get & set file, process or device attributes

5) Communication :-
→ Create, delete communication connections
→ Send & receive messages
→ From client to server

6) Protection :-
→ Maintain control of resources' access
→ Allow & deny user access
→ Get & set permission

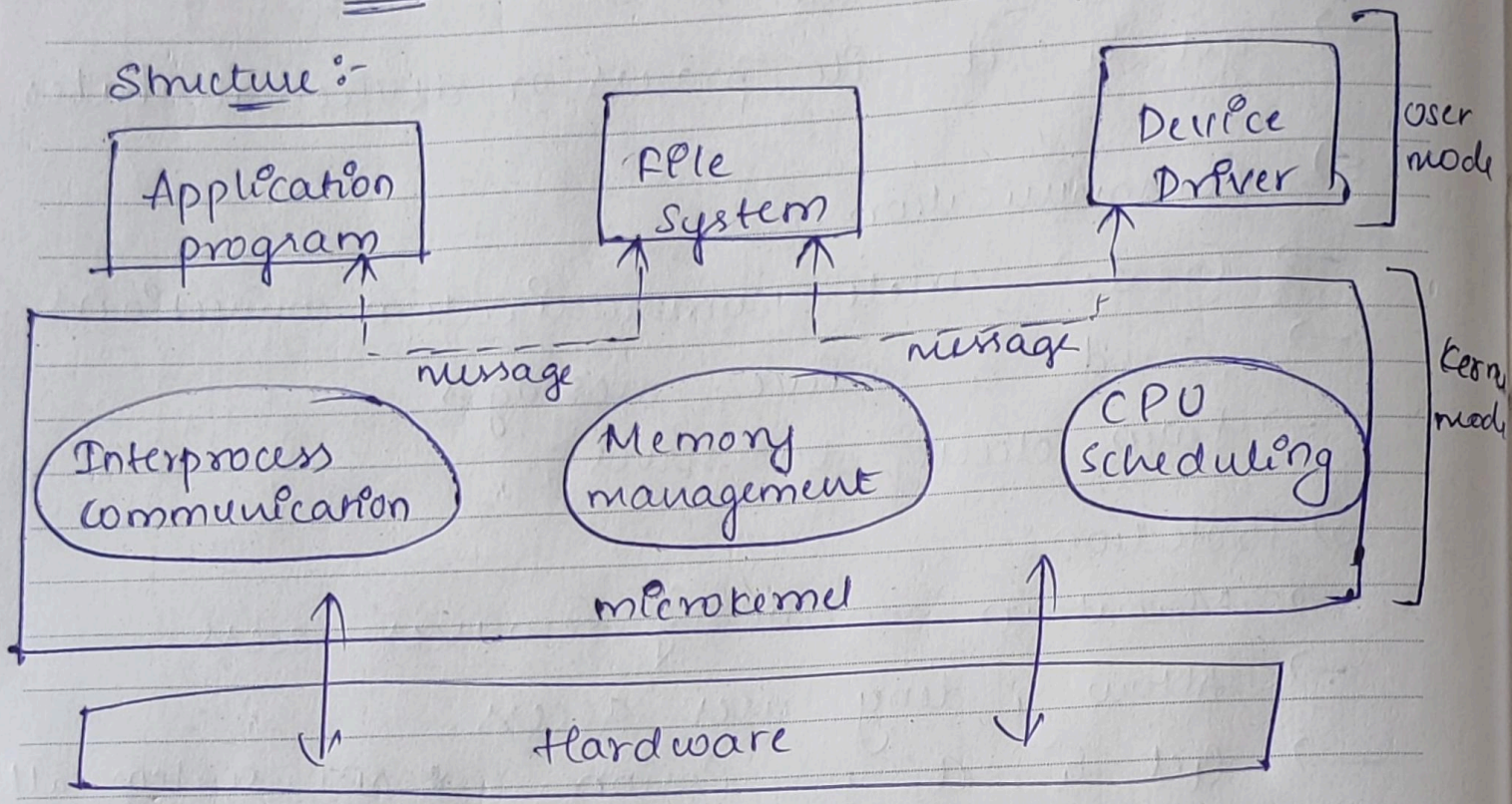
* API - system call - OS Relationship



3] Differentiate between microkernel & monolithic operating system structures.

→ Microkernel :-

Structure :-



→ This structures the OS by removing all the non-essential portions of the kernel & implementing them as system & user program resulting in small kernels.

→ They provide minimal, process & memory management & communication facility.

→ Communication between the components of OS is done using message passing.

Monolithic kernel	Microkernel
1) Kernel size is large	1) Kernel size is small
2) OS is complex to design.	2) OS is easy to design, implement & install.
3) Requested services are provided faster.	3) Requested services are provided slower.
4) Fast execution	4) Slow execution.
5) No Message passing, context switching is required when kernel is running a job.	5) Message passing, context switch is required.
6) All the OS services are provided.	6) Kernel provides IPS, low level device management service.
7) Less code is required	7) More code is required.
8) If the service crashes, the whole system crashes.	8) If the services crashes, working of microkernel won't be affected.
9) Ex:- Microsoft windows, Linux, DOS, etc.	9) Ex:- Mac OS X.

4) Discuss the various services provided by an OS in perspective of program execution.

* Fundamental Goals of OS:-

1) Efficient use of computer resources.
2) User convenience
3) Non-interference in the activities of its users

1) Efficient use of computer resources:-

→ OS ensures the resources are efficiently used.

→ It ensures the efficient use of CPU, memory and I/O devices.

→ It can monitor the use of resources to ensure efficient usage.

→ It uses the policy that ensures efficiency.

2) User convinience:-

→ The OS makes sure that all the needs of the users are met.

→ It must have the ability to execute programs.

→ It has to provide good service & speed response to computation requests

→ It has to be user friendly i.e, easy to use commands.

→ It has to have the facility of concurrent programming.

3) Non-interference :-

→ User finds interferences in the computational activities.

→ Program execution or operation of OS can be disrupted by actions of other persons.

→ The OS prevents interference by providing resources for exclusive use of program & preventing any illegal access to the resources.

→ The OS knows which user files can be accessed by whom.

→ A program is useless until & unless it is given some instructions.

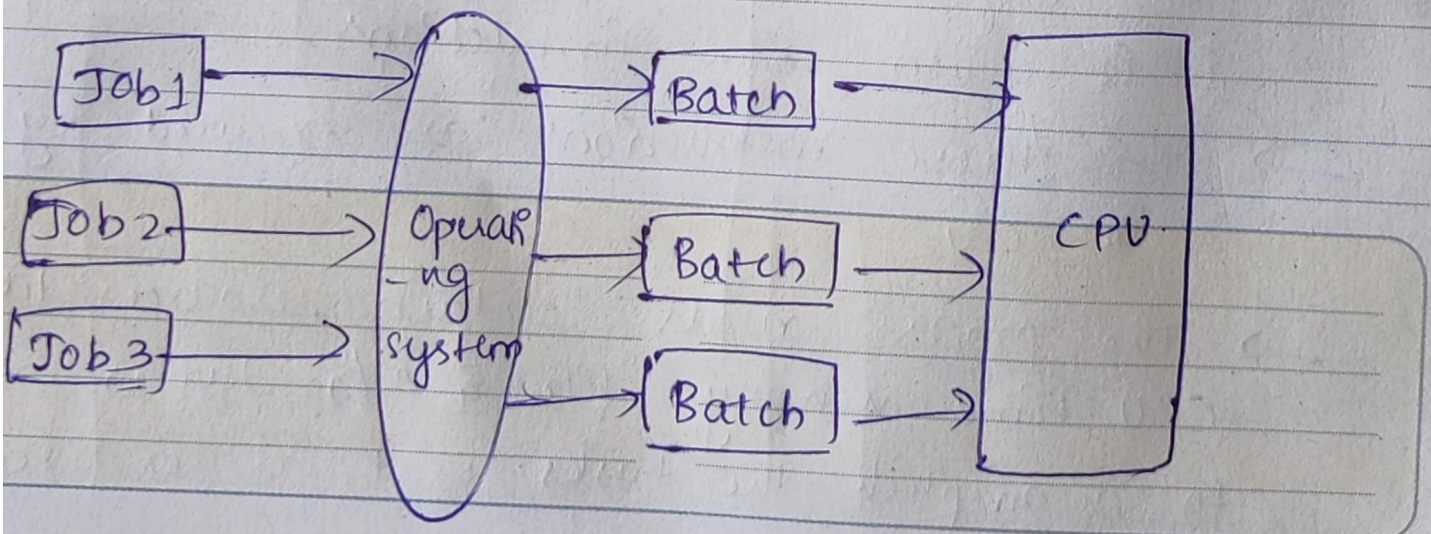
→ These instructions are executed by the CPU.

→ The process needs certain resources like CPU time memory, file & I/O devices to accomplish the tasks that are created.

→ So, the OS ensures that the program execution is done efficiently.
→ The system must be able to load the program, run & execute it.
→ This is ensured by the operating system.

5] Explain Batch, Interactive, time sharing, Real-time & distributed O.S.

→ 1) Batch Operating System:-
→ It is a type of OS that does not interact with the computer directly.
→ It has an operator that is given a job with same requirements and groups them into batches.
→ It is the responsibility of the operator to sort jobs with similar needs.



Advantages :-

1) Multiple users can use batch system at same time.
2) Idle time for batch systems is very less.

Disadvantages:-

1) It is expensive.
2) It is hard to debug.
3) The other jobs will have to wait for an unknown time if any job fails.

ii)