4th Semester Notes

# BS401: Operating System

## BS401: Syllabus

**Operating system Introduction:** Definition, Design Goals, Evolution; Concept of User, job and Resources; Batch processing, Multi-programming, Time sharing; Structure and Functions of Operating System.

**Process Management:** Process states, State Transitions, Process Control Structure, Context Switching, Process Scheduling, Threads.

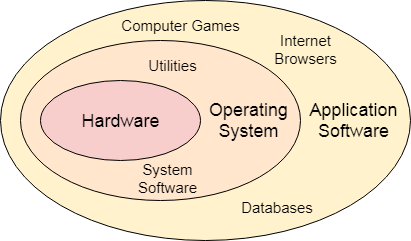
**Memory Management:** Address Binding, Dynamic Loading and Linking Concepts, Logical and Physical Addresses, Contiguous Allocation, Fragmentation, Paging, Segmentation, Combined Systems, Virtual Memory, Demand Paging, Page fault, Page replacement algorithms, Global Vs Local Allocation, Thrashing, Working Set Model, Paging.

**Concurrent Processes:** Process Interaction, Shared Data and Critical Section, Mutual Exclusion, Busy form of waiting, Lock and unlock primitives, Synchronization, Classical Problems of Synchronization, Semaphores, Monitors, Conditional Critical Regions, System Deadlock, Wait for Graph, Deadlock Handling Techniques: Prevention, Avoidance, Detection and Recovery.

**File and Secondary Storage Management:** File Attributes, File Types, File Access Methods, Directory Structure, File System Organization and Mounting, Allocation Methods, Free Space management; Disk Structure, Logical and Physical View, Disk Head Scheduling, Formatting, Swap Management. Protection & Security.

## Unit 01

### Operating system Introduction:

An Operating System (OS) is an interface between a computer user and computer hardware. An operating system is a software which performs all the basic tasks like file management, memory management, process management, handling input and output, and controlling peripheral devices such as disk drives and printers.

Some popular Operating Systems include Linux Operating System, Windows Operating System, VMS, OS/400, AIX, z/OS, etc.

### Definition:

An operating system is a program that acts as an interface between the user and the computer hardware and controls the execution of all kinds of programs.

### Memory Management

Memory management refers to management of Primary Memory or Main Memory. Main memory is a large array of words or bytes where each word or byte has its own address.

Main memory provides a fast storage that can be accessed directly by the CPU. For a program to be executed, it must in the main memory. An Operating System does the following activities for memory management −

* Keeps tracks of primary memory, i.e., what part of it are in use by whom, what part are not in use.
* In multiprogramming, the OS decides which process will get memory when and how much.
* Allocates the memory when a process requests it to do so.
* De-allocates the memory when a process no longer needs it or has been terminated.

### Processor Management

In multiprogramming environment, the OS decides which process gets the processor when and for how much time. This function is called **process scheduling**. An Operating System does the following activities for processor management −

* Keeps tracks of processor and status of process. The program responsible for this task is known as **traffic controller**.
* Allocates the processor (CPU) to a process.
* De-allocates processor when a process is no longer required.

### Device Management

An Operating System manages device communication via their respective drivers. It does the following activities for device management −

* Keeps tracks of all devices. Program responsible for this task is known as the **I/O controller**.
* Decides which process gets the device when and for how much time.
* Allocates the device in the efficient way.
* De-allocates devices.

### File Management

A file system is normally organized into directories for easy navigation and usage. These directories may contain files and other directions.

An Operating System does the following activities for file management −

* Keeps track of information, location, uses, status etc. The collective facilities are often known as **file system**.
* Decides who gets the resources.
* Allocates the resources.
* De-allocates the resources.

### Other Important Activities

Following are some of the important activities that an Operating System performs −

* **Security** − By means of password and similar other techniques, it prevents unauthorized access to programs and data.
* **Control over system performance** − Recording delays between request for a service and response from the system.
* **Job accounting** − Keeping track of time and resources used by various jobs and users.
* **Error detecting aids** − Production of dumps, traces, error messages, and other debugging and error detecting aids.
* **Coordination between other software’s and users** − Coordination and assignment of compilers, interpreters, assemblers and other software to the various users of the computer systems.

### Design & Goals

Operating Systems have become quite complex and multifaceted with the advancement of time. However, that also means it is getting more and more difficult to design operating systems that satisfy all the specifications required these days. There are no complete solutions possible for design problems, but some approaches are more successful than others.

**Design Requirements in Operating System**

The design requirements are quite hard to specify in an operating system. They are basically divided into two parts: User design requirements and System design requirements. Details about these are given as follows −

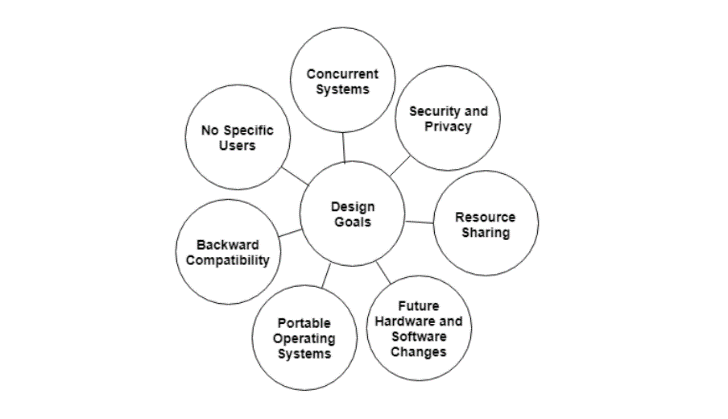
* **User Design Requirements**

The operating system should be convenient, easy to use, reliable, safe and fast according to the users. However, these specifications are not very useful as there is no set method to achieve these goals.

* **System Design Requirements**

The operating system should be easy to design, implement and maintain. These are specifications required by those who create, maintain and operate the operating system. But there is no specific method to achieve these goals as well.

### Design Goals in Operating Systems

There are many design goals that modern operating systems have to fulfil to be considered successful. Some of these are −

### **Concurrent Systems**

Modern operating systems should be able to handle multiple users as well as multiple devices at the same time. This is necessary for the modern multi-core architectures. Because of these specifications, the operating system design can be quite complex and difficult to create.

### **Security and Privacy**

Operating systems should be able to provide security and privacy for a system. This is very important as there are many malicious users who may want to hack into the computer system and steal user programs.

### **Resource Sharing**

The operating system should make sure that resources are shared in a correct manner between multiple user processes. This can get quite complex when multiple users share the same device as well.

### **Future Hardware and Software Changes**

A major design consideration is that the operating system should be able to weather future hardware and software changes and not become obsolete. This is necessary as the operating system being changed again and again is quite a costly process.

### **Portable Operating Systems**

The operating systems should be portable i.e. they should work with different hardware and machines. There may be some speciality operating system that only work on one kind of machine, however, most of them are portable.

### **Backward Compatibility**

The new operating systems created should be compatible with the previous models i.e. they should contain backward compatibility.

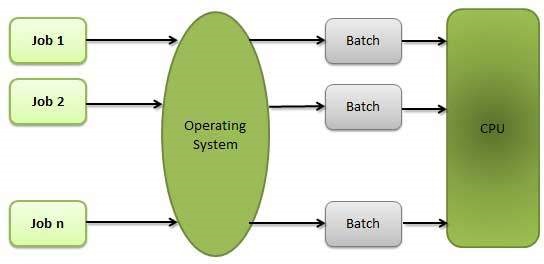
### **No Specific Type of Users**

Operating systems should be developed keeping in mind a general user base so that many users can use them. Even specially developed operating systems that target a single user base contain generality.

### Evolutions:

### Batch Processing:

Batch processing is a technique in which an Operating System collects the programs and data together in a batch before processing starts. An operating system does the following activities related to batch processing −

* The OS defines a job which has predefined sequence of commands, programs and data as a single unit.
* The OS keeps a number a job in memory and executes them without any manual information.
* Jobs are processed in the order of submission, i.e., first come first served fashion.
* When a job completes its execution, its memory is released and the output for the job gets copied into an output spool for later printing or processing.

### **Advantages**

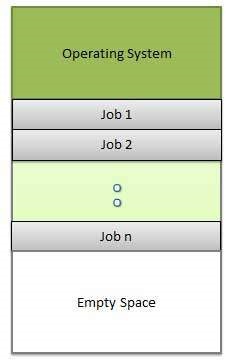
* Batch processing takes much of the work of the operator to the computer.
* Increased performance as a new job gets started as soon as the previous job is finished, without any manual intervention.

### **Disadvantages**

* Difficult to debug program.
* A job could enter an infinite loop.
* Due to lack of protection scheme, one batch job can affect pending jobs.

### Multiprogramming:

Sharing the processor, when two or more programs reside in memory at the same time, is referred as **multiprogramming**. Multiprogramming assumes a single shared processor. Multiprogramming increases CPU utilization by organizing jobs so that the CPU always has one to execute.

The following figure shows the memory layout for a multiprogramming system.

An OS does the following activities related to multiprogramming.

* The operating system keeps several jobs in memory at a time.
* This set of jobs is a subset of the jobs kept in the job pool.
* The operating system picks and begins to execute one of the jobs in the memory.
* Multiprogramming operating systems monitor the state of all active programs and system resources using memory management programs to ensures that the CPU is never idle, unless there are no jobs to process.

### **Advantages**

* High and efficient CPU utilization.
* User feels that many programs are allotted CPU almost simultaneously.

### **Disadvantages**

* CPU scheduling is required.
* To accommodate many jobs in memory, memory management is required.

### Interactivity

Interactivity refers to the ability of users to interact with a computer system. An Operating system does the following activities related to interactivity −

* Provides the user an interface to interact with the system.
* Manages input devices to take inputs from the user. For example, keyboard.
* Manages output devices to show outputs to the user. For example, Monitor.

The response time of the OS needs to be short, since the user submits and waits for the result.

### Time-Sharing Operating Systems:

Time-sharing enables many people, located at various terminals, to use a particular computer system at the same time. Multitasking or Time-Sharing Systems is a logical extension of multiprogramming. Processor’s time is shared among multiple users simultaneously is termed as time-sharing.

The main difference between Time-Sharing Systems and Multiprogramming Batch Systems is that in case of Multiprogramming batch systems, the objective is to maximize processor use, whereas in Time-Sharing Systems, the objective is to minimize response time.

Multiple jobs are implemented by the CPU by switching between them, but the switches occur so frequently. So, the user can receive an immediate response. For an example, in a transaction processing, the processor executes each user program in a short burst or quantum of computation, i.e.; if n users are present, then each user can get a time quantum. Whenever the user submits the command, the response time is in few seconds at most.

An operating system uses CPU scheduling and multiprogramming to provide each user with a small portion of a time. Computer systems which were designed primarily as batch systems have been modified to time-sharing systems.

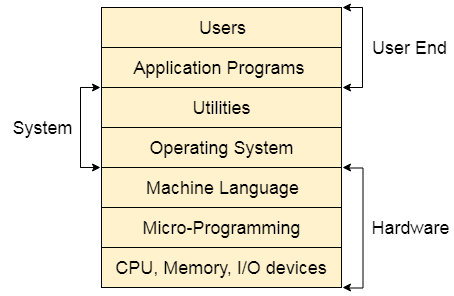
Advantages of Timesharing operating systems are −

* It provides the advantage of quick response.
* This type of operating system avoids duplication of software.
* It reduces CPU idle time.

Disadvantages of Time-sharing operating systems are −

* Time sharing has problem of reliability.
* Question of security and integrity of user programs and data can be raised.
* Problem of data communication occurs.

### Structure & Function of Operating System

**Structure of a Computer System**

A Computer System consists of:

* **Users** (people who are using the computer)
* **Application Programs** (Compilers, Databases, Games, Video player, Browsers, etc.)
* **System Programs** (Shells, Editors, Compilers, etc.)
* **Operating System** (A special program which acts as an interface between user and hardware)
* **Hardware** (CPU, Disks, Memory, etc.)

**Following are some of important functions of an operating System.**

1. Process Management
2. Process Synchronization
3. Memory Management
4. File Management
5. Device Management
6. CPU Scheduling
7. Security
8. Error detecting aids
9. Control over system performance
10. Coordination between other software and users

## Unit 02

### Process Management:

## Unit 03

## Unit 04

## Unit 05