SRINIVAS UNIVERSITY

INSTITUTE OF COMPUTER SCIENCE & INFORMATION SCIENCES

CITY CAMPUS, PANDESHWAR, MANGALORE - 575 001

BACKGROUND STUDY MATERIAL

OPERATING SYATEM

B.C.A III SEMESTER

Compiled by

Faculty

OPERATING SYSTEM TEACHING PLAN

PERIOD	TOPIC
Module 1	
1	Introduction - Operating System Definition, Simple batch system
2	Multi programmed batched system, Time sharing system
3	Real- time system, System components
4	Operating system services, Process Management - Process Concept
5	Process Scheduling
6	Types of Schedulers
7	Co-Operating Process
8	Threads
Module 2	
9	CPU Scheduling - Factors affecting the scheduling
10	Preemptive and non-preemptive scheduling
11	Scheduling Criteria
12	FCFS
13	SJF Priority
14	Round robin
15	Multi-level Scheduling
16	multi-level feedback queue Algorithm
Module 3	
17	Deadlocks - deadlock, Necessary and Sufficient Condition for a deadlock state
18	resource Allocation Graph, Wait-for Graph
19	Dead Lock Prevention, Deadlock Avoidance
20	Recovery from deadlock, Memory Management - Logical and Physical Address Space
21	Swapping the Pages
22	Contiguous Allocation (Memory Allocation; Fragmentation)
23	Free Space Management

24	Segmentation
Module 4	
25	Virtual Memory Management - Demand Paging, Page Replacement
26	Page Replacement Algorithms, Allocation of Frames (Equal and Proportional Allocation)
27	Thrashing(concept), File Systems - File Concept
28	Access Methods
29	Directory Structures
30	Allocation Methods
31	Free Space Management
32	Protection of File System
Module 5	
33	Linux - An Introduction, reason for its popularity
34	Directory oriented command, wild card characters
35	File oriented commands
36	File Access Permissions
37	Process oriented commands
38	Communication oriented commands purpose commands
39	general Purpose Commands
40	Pipe and Filters related commands

OPERATING SYSTEM SYLLABUS

Paper : 18BCASD34 **Theory/Week:**4 Hours **Credits:**4	**Operating Systems**	**Hours:** 40 **IA :** 50 **Exam:** 50

Unit – I **8hrs**

Introduction - Operating System Definition, Simple batch system, Multi programmed batched system, Time-sharing system, Real- time system, System components, and Operating system services. **Process Management -** Process Concept, Process Scheduling, Types of Schedulers, Co-Operating Process, Threads.

Unit – II **8hrs**

CPU Scheduling - Factors affecting the scheduling, Pre-emptive and non-pre-emptive scheduling, Scheduling Criteria, FCFS, SJF Priority, Round robin, Multi-level Scheduling, multi-level feedback queue Algorithm.

Unit – III **8hrs**

Deadlocks - deadlock, Necessary and Sufficient Condition for a deadlock state, resource Allocation Graph, Wait-for Graph, Dead Lock Prevention, Deadlock Avoidance, Recovery from deadlock**. Memory Management -** Logical and Physical Address Space, Swapping the Pages, Contiguous Allocation (Memory Allocation , Fragmentation),Paging, Segmentation

Unit – IV **8hrs**

Virtual Memory Management - Demand Paging, Page Replacement, Page Replacement Algorithms, Allocation of Frames (Equal and Proportional Allocation), Thrashing(concept) **File Systems -** File Concept, Access Methods, Directory Structures, Allocation Methods, Free Space Management, Protection of File System.

Unit – V **8hrs**

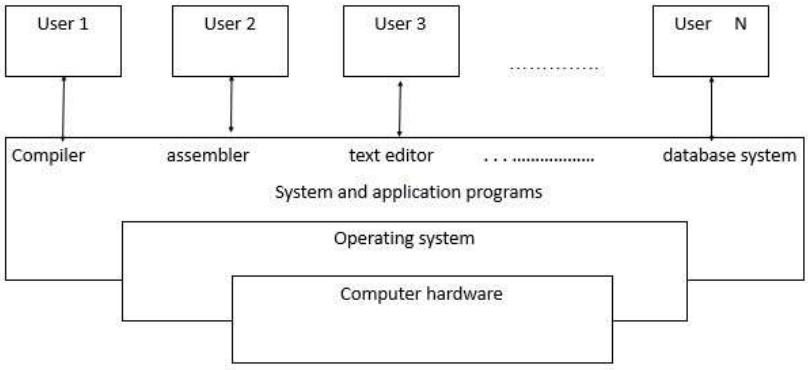
Linux - An Introduction, reason for its popularity, Directory oriented command, wild card characters, File oriented commands, File Access Permissions, Process oriented commands, Communication oriented commands purpose commands, general Purpose Commands, Pipe and Filters related commands.

Reference Books

1	Abraham Silberschartz and Peter Galvin, Operating System Concepts 6th edition, TMH.
2	B Mohammed Ibrahim, Linux: A Practical Approach, Firewall Media, 2009.

CHAPTER 1
INTRODUCTION

An operating system is a program that manages the computer hardware. It also provides a basis for application programs and acts as an intermediary between the computer user and the computer hardware. An amazing aspect of operating systems is how varied they are in accomplishing these tasks. Mainframe operating systems are designed primarily to optimize utilization of hardware. Personal computer (PC) operating systems support complex games, business applications, and everything in between. Operating systems for hand held computers are designed to provide an environment in which a user can easily interface with the computer to execute programs.



A computer system can be divided roughly into four components: the hardware, the operating system, the application programs, and the users. The hardware - the central processing unit (CPU), the memory, and the input/output (I/O) devices - provides the basic computing resources for the system. The application programs - such as word processors, spreadsheets, compilers, and Web browsers - define the ways in which these resources are used to solve users' computing problems.

The operating system controls the hardware and coordinates its use among the various application programs for the various users. We can also view a computer system as consisting of hardware, software, and data. The operating system provides the means for proper use of these resources in the operation of the computer system. An operating system is similar to a government. Like a government, it performs no useful function by itself. It simply provides an environment within which other programs can do useful work.

An OS can be explored in two viewpoints: User View and System View.

USER VIEW

The user view of a computer varies by the interfaces being used. In the view where the user sits in front of the PC, performance is important to the user but does not matter about the system being idle or waiting for the slow I/O speed of the user. Here, the OS is designed mostly for ease of use and performance. Some users may be at terminals connected to mainframes. Others may access the same computer through other terminals. The users here share resources and exchange information. The OS in this case is designed to maximize resource utilization - to ensure that CPU time, memory and I/O are used efficiently. Some users may be at workstations connected to networks of other work stations and servers. These users have dedicated resources and at the same time share resources. The OS in this case is designed to compromise between individual usability and resource utilization. Some computers have little or no user view. For example, embedded computers in home devices and automobiles may have numeric keypads and may turn indicator lights on or off to show status, but they and their operating systems are designed primarily to run without user intervention.

SYSTEM VIEW

From computers point of view, an OS is the most intimate with the hardware. We can view OS as a resource allocator. A computer has many resources that may be required to solve a problem - CPU time, memory space and I/O devices. The OS acts as a manager of these resources. It decides on how to allocate these resources to specific programs and users so that it can operate the computer system efficiently. A slightly different view is the need to control the various I/O devices and user programs. An OS acts as a control program. A control program manages the execution of various user programs to prevent errors and improper use of computer. Thus, the common function of controlling and allocating the resources are brought together into one piece of software called Operating Systems.

An operating system (OS) exploits the hardware resources of one or more processors to provide a set of services to system users. The OS also manages secondary memory and I/O (input/output) devices on behalf of its users. At a top level, a computer consists of processor, memory, and I/O components, with one or more modules of each type. These components are interconnected in some fashion to achieve the main function of the computer, which is to execute programs. Thus, there are four main structural elements:

1. **Processor:** Controls the operation of the computer and performs its data processing functions. When there is only one processor, it is often referred to as the central processing unit (CPU).
2. **Main memory:** Stores data and programs. This memory is typically volatile; that is, when the computer is shut down, the contents of the memory are lost. In contrast, the contents of disk memory are retained even when the computer system is shut down. Main memory is also referred to as real memory or primary memory
3. **I/O modules:** Move data between the computer and its external environment. The external environment consists of a variety of devices, including secondary memory devices (e.g., disks), communications equipment, and terminals.
4. **System bus:** Provides for communication among processors, main memory, and I/O modules.

COMPUTER SYSTEM ARCHITECTURE

A computer system can be organized in a number of different ways, which we can categorize roughly according to the number of general-purpose processors used.

- Single-Processor Systems
- Multi-Processor Systems

1**. Single-Processor Systems**

If there is only one general-purpose CPU, then the system is a single-processor system. On a single-processor system, there is one main CPU capable of executing a general-purpose instruction set, including instructions from user processes. Almost all systems have other special-purpose processors as well. They may come in the form of device-specific processors, such as disk, keyboard, and graphics controllers; or, on mainframes, they may come in the form of more general-purpose processors, such as I/O processors that move data rapidly among the components of the system. All of these special-purpose processors run a limited instruction set and do not run user processes. Sometimes they are managed by the operating system, in that the operating system sends them information about their next task and monitors their status. For example, a disk-controller microprocessor receives a sequence of requests from the main CPU and implements its own disk queue and scheduling algorithm. This arrangement relieves the main CPU of the overhead of disk scheduling. PCs contain a microprocessor in the keyboard to convert the keystrokes into codes to be sent to the CPU. In

other systems or circumstances, special-purpose processors are low-level components built into the hardware. The operating system cannot communicate with these processors; they do their jobs autonomously.

2. Multi-Processor System

Systems have two or more processors in close communication, sharing the computer bus and sometimes the clock, memory, and peripheral devices.

Multiprocessor systems have three main advantages:

1. **Increased throughput.**

By increasing the number of processors, we expect to get more work done in less time. The speed-up ratio with N processors is not N, however; rather, it is less than N. When multiple processors cooperate on a task, a certain amount of overhead is incurred in keeping all the parts working correctly. This overhead, plus contention for shared resources, lowers the expected ain from additional processors. Similarly, N programmers working closely together do not produce N times the amount of work a single programmer would produce.

2. **Economy of scale.**

Multiprocessor systems can cost less than equivalent multiple single processor systems, because they can share peripherals, mass storage, and power supplies. If several programs operate on the same set of data, it is cheaper to store those data on one disk and to have all the processors share them than to have many computers with local disks and many copies of the data.

3. **Increased reliability.**

If functions can be distributed properly among several processors, then the failure of one processor will not halt the system, only slow it down. If we have ten processors and one fails, then each of the remaining nine processors can pick up a share of the work of the failed processor. Thus, the entire system runs only 10 percent slower, rather than failing altogether.

Difference between Single Processor Systems and Multiprocessor Systems

Parameter	Single Processor Systems	Multiprocessor Systems
Description	If a System contains only one processor for processing than it is called single processor system.	If a System contains two or more than two processors for processing than it is called multiprocessor system.

Use of Co-Processors	Yes, Single Processors uses multiple Controllers that are Designed to handle special tasks and can execute limited instruction sets. Ex: DMA Controller, North/South Bridge.	In Multi Processor Systems Two Types of approaches are Used: 1)Symmetric Multiprocessing(SMP) 2) Asymmetric Multiprocessing In Asymmetric Multiprocessing one Processor works as Master and Second Processor act as Slave And in Symmetric Multiprocessing each processor performs all the tasks within the operating system.
Throughput	Throughput of Single Processor Systems is less than Multiprocessor Systems Because every task is performed by the same processor.	Throughput of Multiprocessor systems is greater than single processor systems. If a System Contains N Processors Than its throughput will be slightly less than N because synchronization must be maintained between two processors and they also share resource which increases certain amount of overhead.
Cost Economic	Single Processor Systems cost more because each processor requires separate resources. i.e. Mass Storage, Peripherals, Power Supplies etc.	Multiprocessor Systems cost less than equivalent multiple single processor systems because they uses same resources on sharing basis.

Design Process	It is Easy to design Single Processor Systems.	It is difficult to design Multi Processor Systems because Synchronization must be maintained between processors otherwise it may result in overloading of one processor and another processor may remain idle on the same time.
Reliability	Less reliable because failure in one processor will result in failure of entire system.	More reliable because failure of one processor does not halt the entire system but only speed will be slow down.
Examples	Most of Modern PCs.	Blade Servers.

TYPES OF OPERATING SYSTEM

An Operating System performs all the basic tasks like managing file, process, and memory. Thus operating system acts as manager of all the resources, i.e. resource manager. Thus operating system becomes an interface between user and machine. Many different types of operating systems are involved till date. The operating systems are improved in terms of their capabilities. The modern days operating systems allows multiple user to carry out multiple tasks simultaneously. Based on their capabilities and the types of application supported, the operating systems can be divided into following four major categories.

1. Batch processing operating systems
2. Multi programming operating systems
3. Time sharing operating systems
4. Real time operating systems

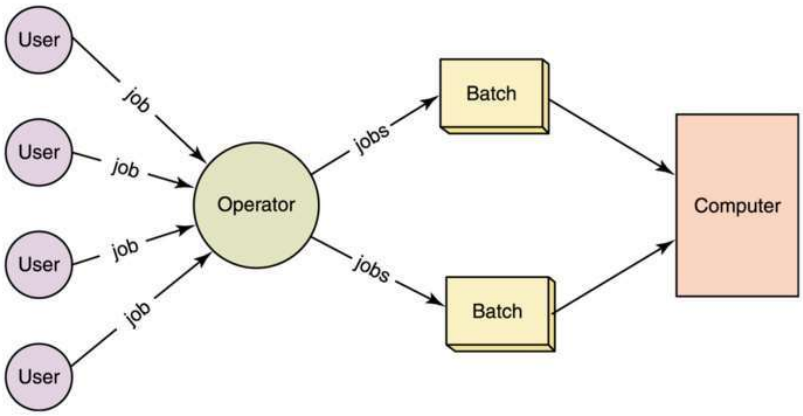
1. **Batch processing operating systems**

Jobs that can run without end user interaction, or can be scheduled to run as resources permit, are called batch jobs. Batch processing is basically for those frequently used programs that can be run with minimal human interaction. A program that reads a large file and generates a report, for example, is considered to be a batch job.

Early computers were enormous machines run from a console. The common input devices were card readers and tape drives. The common output devices were line printers,

tape drives and card punches. The user did not interact directly with the computer. Rather, the user prepared a job which consisted of program, data and some control information and submitted to the computer operator. The job was usually in the form of punch cards. After some later time the output appeared which consisted of the result of the program as well as the dump of the final memory and register contents for debugging.

The OS in early computers was simple. The task of OS was to transfer control automatically from one job to the next. The OS was always resident in the memory. To speed up processing, operators batched together jobs with similar needs and ran them through the computer as a group. Thus, the programmers would leave their program with the operator. The operator would sort the programs into batches and as the computers became available, would run each batch. The output of each batch would be sent to the appropriate programmer. In this execution environment, the CPU is often idle because the speed of mechanical devices is slower than electronic devices. With the advent of disk technology, the OS kept all the jobs on the disk rather than in serial card reader. With direct access to several jobs on the disk, the CPU would perform job scheduling to use resources and perform tasks efficiently.



Batch operating system users do not interact with the computer directly. Each job is prepared by each user on an off-line device like punch cards and submits it to the computer operator. To increase the processing speed, jobs with similar needs are batched together and run as a group. After completing the coding for the programs the programmers hand over their programs to the operator and the operator then analyze the programs having similar requirements and then divide them into batches.

Advantages of Batch Systems

- Repeated jobs are done fast in batch systems without user interaction.
- You don't need special hardware and system support to input data in batch systems.

- Best for large organizations but small organizations can also benefit from it.
- Batch systems can work offline so it makes less stress on processor.
- Processor consumes good time while processing that mean it knows which job to process next. In real time systems we don't have expectation time of how long the job is and what is estimated time to complete it. But in batch systems the processor knows how long the job is as it is queued.
- Sharing of batch system for multiple users.
- The idle time batch system is very less.
- You can assign specific time for the batch jobs so when the computer is idle it starts processing the batch jobs i.e. at night or any free time.
- The batch systems can manage large repeated work easily.

Disadvantages of Batch Systems

1. **Sequential execution:** This is one of the major disadvantages of Batch Systems. Jobs in a batch are always executed sequentially. For example, if there are 4 jobs in one batch, then they are always executed one by one and the output is obtained only once all 4 jobs are completed. so Difficult to provide the desired priority.
2. **Starvation:** Different jobs in a single batch might take different amounts of time in execution. This might lead to starvation of some jobs. Suppose there are 4 jobs in a batch and the first job takes too long to execute then the other three jobs in the same batch will have to wait for long until the first one is completed.
3. **No interaction between job and user**: Once a batch is submitted to the computer, the user is no longer able to interact with any of the jobs. Suppose there is a job which requires the user to give the input data during runtime. Now, He must wait until all the jobs in that batch are completed. So, the overall execution time is increased a lot.

Ex: Payroll System, Bank Statements etc.

2. Multi programming operating systems

The most important aspect of job scheduling is the ability to multi program. A single user cannot keep either the CPU or I/O devices busy all the time. Multiprogramming increases CPU utilization by organizing jobs so that CPU always busy with executing one job. The idea is as follows - the OS keeps several jobs in memory simultaneously. The set of jobs is a subset of jobs in the job pool. The as picks and executes one of the jobs in the memory. The job may have to wait for some task such as an I/O operation to complete. In non-multi programmed environment, the CPU would sit idle. In multi-programmed

environment the OS switches to another job. When that job needs to wait, the CPU switches to another job and so on. Eventually the first job finishes waiting and gets the CPU. As long as there is at least one job to execute the CPU never sits idle. Multi-programming is the first instance where the OS has to make decisions for the user.

All the jobs that enter the system are kept in a job pool. A pool is a set of all processes residing on the disk awaiting allocation of main memory. If several jobs are ready to be brought into memory and if there is not enough space for all of them, then the system must choose from among them. This decision making capability is called job scheduling. When the OS selects a job from job pool, it loads that job into memory for execution. If several jobs re ready to run at the same time, the system must choose among them. This decision making capability of as is called CPU scheduling.

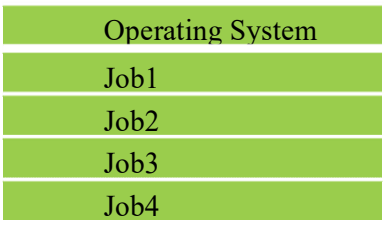


Figure: Memory Layout for a multiprogramming System

Advantages of multiprogramming systems

- CPU is used most of time and never become idle
- The system looks fast as all the tasks runs in parallel
- Short time jobs are completed faster than long time jobs
- Multiprogramming systems support multiply users
- Resources are used nicely
- Total read time taken to execute program/job decreases
- Response time is shorter
- In some applications multiple tasks are running and multiprogramming systems better handle these type of applications

Disadvantages of multiprogramming systems

- It is difficult to program a system because of complicated schedule handling
- Tracking all tasks/processes is sometimes difficult to handle
- Due to high load of tasks, long time jobs have to wait long

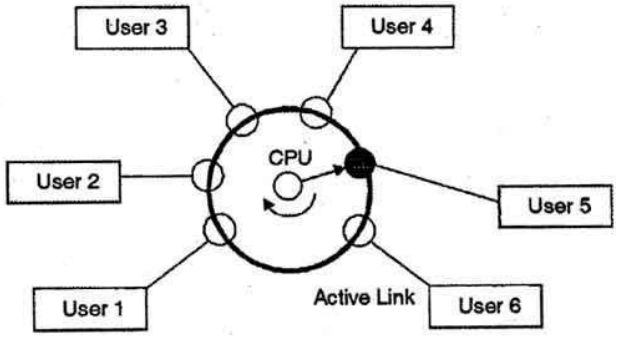
Ex: A computer running excel and chrome browser simultaneously.

3. Time sharing operating systems

A time-sharing system allows many users to share the computer resources simultaneously. In other words, time-sharing refers to the allocation of computer resources in time slots to several programs simultaneously. For example a mainframe_computer that has many users logged on to it. Each user uses the resources of the mainframe -i.e. memory, CPU etc. The users feel that they are exclusive user of the CPU, even though this is not possible with one CPU i.e. shared among different users.

The time sharing systems were developed to provide an interactive use of the computer system. A time shared system uses CPU scheduling and multiprogramming to provide each user with a small portion of a time-shared computer. It allows many users to share the computer resources simultaneously. As the system switches rapidly from one user to the other, a short time slot is given to each user for their executions.

The time-sharing system provides the direct access to a large number of users where CPU time is divided among all the users on scheduled basis. The OS allocates a set of time to each user. When this time is expired, it passes control to the next user on the system. The time allowed is extremely small and the users are given the impression that they each have their own CPU and they are the sole owner of the CPU. This short period of time during that a user gets attention of the CPU; is known as a time slice or a quantum. The concept of time-sharing system is shown in figure.



In above figure the user 5 is active but user 1, user 2, user 3, and user 4 are in waiting state whereas user 6 is in ready status. As soon as the time slice of user 5 is completed, the control moves on to the next ready user i.e. user 6. In this state user 2, user 3, user 4, and user 5 are in waiting state and user 1 is in ready state. The process continues in the same way and so on.

The time-shared systems are more complex than the multi-programming systems. In timeshared systems multiple processes are managed simultaneously which requires an adequate management of main memory so that the processes can be swapped in or swapped

out within a short time. Note that the term 'Time Sharing' is no longer commonly used; 'Multitasking System' has replaced it.

Advantages:

- Duplication of software is less probable
- Each task is given equal importance
- The CPU idle time can be decreased

Disadvantages:

- Problem of reliability
- Care of security and integrity is to be taken of user data and programs
- There is a problem in data communication

Ex: a mainframe computer that has many users logged on to it. Each user uses the resources of the mainframe -i.e. memory, CPU etc.

4. Real time operating systems

A real time system is used when rigid time requirements have been placed on the operation of the processor or flow of data. It is used in dedicated applications as a control device. Sensors bring data to a computer. The computer must analyze the data and possibly adjust controls to modify the sensor input. . Examples are medical imaging systems, industrial control systems. . A real time system has well defined time constraints. Processing must be done within the defined constraints or the system will fail. Real time systems can be classified as : a) hard real time systems and b) soft real time systems. A hard real time system guarantees that the critical tasks are completed on time. All delays in the system should be bounded. Delays could be due to retrieval of stored data or time for as to finish any request made of it. A soft real time system is less restrictive. A critical real time tasks gets priority over other tasks and retains that priority until it completes.

Advantages:

- **Error free:** Real time operating systems are error-free. So there is no chance of getting an error when we are performing the task.
- **Maximum Consumption:** we can achieve maximum consumption by using RTOS. It keeps all the devices active and produces more output by making use of all the resources.
- **Embedded system usage:** As the programs of the RTOS are of small size. So, we make use of RTOS in embedded systems.

- **Task shifting:** In RTOS, shifting time of the tasks is very small.
- **24/7 performance:** RTOS is best used for those applications which run constantly or you can say that which runs for 24 hours and 7 days in a week. This is possible because of fewer shifting of tasks so that it can give maximum output.
- **Application focus:** In RTOS, a very few tasks are managed so that exact results can be given by the system. It gives very less importance to those applications which are in waiting state.

Disadvantages

- **Limited tasks:** RTOS can run very limited tasks simultaneously and it concentrates on only those applications which contain error so that it can avoid them. Due to this, some tasks have to wait for unlimited time.
- **Low multi-tasking:** As RTOS is the system which concentrates on few tasks. So sometimes it is difficult for these systems to do multi-tasking.
- **Expensive:** As stated earlier, a lot of resources are used by RTOS which makes it expensive.
- **Low priority of tasks:** The tasks which has the low priority has to wait for a long time as the RTOS maintain the accuracy of the current programs which are under execution.
- **Complex algorithms:** It makes use of some complex algorithms so that the system can give the desired output. These complex algorithms are difficult to understand.
- **Use heavy system resources:** Real time operating system makes use of a lot of resources. Sometimes it is not good for the system. It seems to be expensive also.
- **Not easy to program:** Sometime the programmer has to write the complex programs for the efficient working of the RTOS which is not an easy task.
- **Other factors:** Some other factors like error handling, memory management and CPU are also needed to be considered.

Example: Airlines reservation system, Air traffic control system, Systems that provide immediate updating, Systems that provide up to the minute information on stock prices, Defence application systems like as RADAR.

SYSTEM COMPONENTS

A system component is a process, program, utility, or another part of a computer's operating system that helps to manage different areas of the computer. Not to be confused

with a hardware component, a system component is similar to a computer program, but is not something an end-user directly interact with when using a computer. There are multiple system components at work in a computer operating system, each serving a specific function. Together, they allow the operating system and computer to function correctly and efficiently.

1. **Process Management**

A process can be thought of a program in execution. Example: word processing application runs by an individual, a system task of sending output to a printer. A process needs certain resources - CPU time, memory, files and I/O devices to complete a task. These resources are given to it when the process is created or while it is running. A program is a passive entity such as contents of a file stored on a disk. A process is an active entity with a program counter specifying the next instruction to be executed. The execution of a process is sequential. The CPU executes one instruction of a process after another until the process is complete. Thus, a system consists of a collection of processes that include as processes and user processes.

The operating system is responsible for the following activities in connection with processes managed.

- The creation and deletion of both user and system processes
- The suspension are resumption of processes.
- The provision of mechanisms for process synchronization
- The provision of mechanisms for deadlock handling.
- Providing mechanism for process communication

2. **Main memory management**

Memory is central to the operation of a modern computer system. Memory is a large array of words or bytes, each with its own address. Interaction is achieved through a sequence of reads or writes of specific memory address. The CPU fetches from and stores in memory. In order for a program to be executed it must be mapped to absolute addresses and loaded in to memory. As the program executes, it accesses program instructions and data from memory by generating these absolute is declared available, and the next program may be loaded and executed. In order to improve both the utilization of CPU and the speed of the computer's response to its users, several processes must be kept in memory. There are many different algorithms depends on the particular situation. Selection of a memory management scheme for a specific system depends upon many factor, but especially upon the hardware design of the system. Each algorithm requires its own hardware support.

The operating system is responsible for the following activities in connection with memory management.

- Keep track of which parts of memory are currently being used and by whom.
- Decide which processes are to be loaded into memory when memory space becomes available.
- Allocate and deallocate memory space as needed.

3. **Secondary Storage Management**

The main purpose of a computer system is to execute programs. These programs, together with the data they access, must be in main memory during execution. Since the main memory is too small to permanently accommodate all data and program, the computer system must provide secondary storage to backup main memory. Most modem computer systems use disks as the primary on-line storage of information, of both programs and data. Most programs, like compilers, assemblers, sort routines, editors, formatters, and so on, are stored on the disk until loaded into memory, and then use the disk as both the source and destination of their processing. Hence, the proper management of disk storage is of central importance to a computer system. There are few alternatives. Magnetic tape systems are generally too slow. In addition, they are limited to sequential access. Thus, tapes are more suited for storing infrequently used files, where speed is not a primary concern.

The operating system is responsible for the following activities in connection with disk management

- Free space management
- Storage allocation
- Disk scheduling.

4. **I/o system**

One of the purposes of an operating system is to hide the peculiarities of specific hardware devices from the user. For example, in Unix, the peculiarities of I/O devices are hidden from the bulk of the operating system itself by the I/O system. The I/O system consists of:

- A buffer caching system
- A general device driver code
- Drivers for specific hardware devices.

5. **File management**

File management is one of the most visible services of an operating system. Computers can store information in several different physical forms; magnetic tape, disk, and drum are the most common forms. Each of these devices has it own characteristics and physical organization. For convenient use of the computer system, the operating system provides a uniform logical view of information storage. The operating system abstracts from the physical properties of its storage devices to define a logical storage unit, the file. Files are mapped, by the operating system, onto physical devices.

A file is a collection of related information defined by its creator. Commonly, files represent programs (both source and object forms) and data. Data files may be numeric, alphabetic or alphanumeric. Files may be free-form, such as text files, or may be rigidly formatted. In general a files is a sequence of bits, bytes, lines or records whose meaning is defined by its creator and user. It is a very general concept. The operating system implements the abstract concept of the file by managing mass storage device, such as types and disks. Also files are normally organized into directories to ease their use. Finally, when multiple users have access to files, it may be desirable to control by whom and in what ways files may be accessed.

The operating system is responsible for the following activities in connection with file management:

- The creation and deletion of files
- The creation and deletion of directory
- The support of primitives for manipulating files and directories
- The mapping of files onto disk storage.
- Backup of files on stable (non volatile) storage.

6. **Protection system**

The various processes in an operating system must be protected from each others activities. For that purpose, various mechanisms which can be used to ensure that the files, memory segment, CPU and other resources can be operated on only by those processes that have gained proper authorization from the operating system. For example, memory addressing hardware ensure that a process can only execute within its own address space. The timer ensure that no process can gain control of the CPU without relinquishing it. Finally, no process is allowed to do it's own I/O, to protect the integrity of the various peripheral devices.

Protection refers to a mechanism for controlling the access of programs, processes, or users to the resources defined by a computer controls to be imposed, together with some means of enforcement. Protection can improve reliability by detecting latent errors at the interfaces between component subsystems. Early detection of interface errors can often prevent contamination of a healthy subsystem by a subsystem that is malfunctioning. An unprotected resource cannot defend against use (or misuse) by an unauthorized or incompetent user.

7. **Command line interpreter**

Command interpreter is an interface between the user and OS. Many commands are given to as by control statements. When a new job is started in a batch or when a user logs into time shared system, a program that reads and interprets control statements are execute automatically. This program is called command line interpreter or control card interpreter and is often known as shell. Its function is simple - to get the next command statement and execute it.

OPERATING SYSTEM SERVICES

An OS provides an environment for the execution of program. It provides certain services to the program and the user of the program. The services are provided for the convenience of the programmer to make programming task easier. The services provided by the as are

- **Program execution:** The system must be able to load a program into memory and to run the program. The program must be able to end its execution either normally or abnormally (by indicating an error)
- **I/O operations:** A running program may require I/O. This I/O may be a file or an I/O device. Users cannot control I/O devices directly. Hence, an OS must provide a means for I/O operation.
- **File system manipulation:** The OS must provide means to manage and work with file.
- **Communication:** One process needs to exchange information with another process. This could take place between the processes within the same computer or between different computers tied through a network. Communication can be implemented by shared memory or by message passing, in which packets of information are moved between processes by OS