CONTENTS

S.NO	CHAPTER NAME	PAGE NO
1	INTRODUCTION	1-10
2	PROCESS MANAGEMENT	11-24
3	MEMORY MANAGEMENT	25-37
4	DEVICE MANAGEMENT	38-47
5	DEAD LOCKS	48-58
6	FILE MANAGEMENT	59-75
7	SYSTEM PROGRAMMING	76-80

UNIT-1
INTRODUCTION

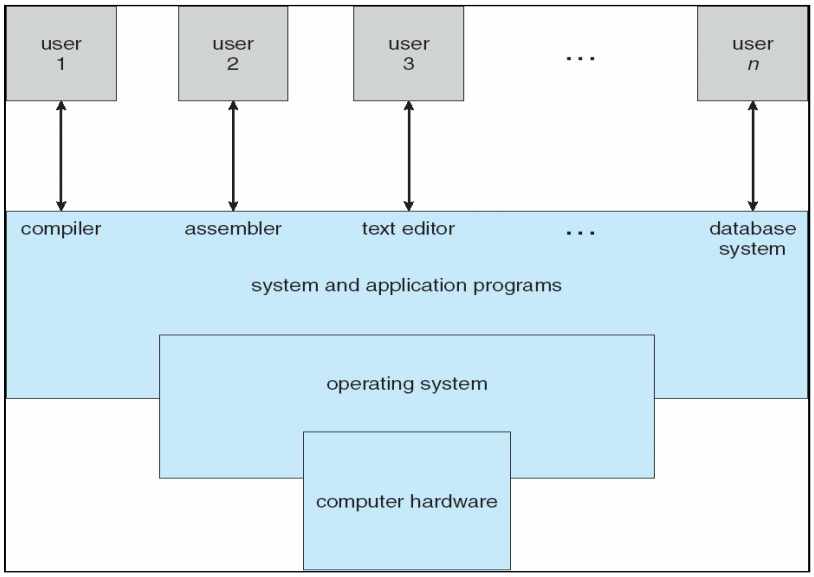
INTRODUCTION:

- Operating system is a system software that acts as an intermediary between the user of a computer and computer hardware.
- It is considered as the brain of the computer.
- It controls the internal activities of the comp. hardware and provides the user interface.
- This interface enables a user to utilize the hardware resources very efficiently.
- It is the first program that gets loaded into the computer memory through the process called "booting".

COMPONENTS OF COMPUTER SYSTEM:

In general, we can divide a computer system into the following four components.

- Hardware
- Operating system
- Application programs
- Users



- As we can see in the figure the user interacts with the application programs.
- The application program do not access the hardware resources directly.
- HARDWARE resources include I/O devices, primary memory, secondary memory(hard disk, floppy disk etc.)and the microprocessor.
- So the operating System is required to access and use these resources.
- The application programs are programmed in such a way that they can easily communicate with the resources.
- An operating System is the first program that is loaded into into the computers main memory, when acomputer is switched on.
- Some popular operating System are windows9x(95,98), Linux, Unix, Windows xp. vista etc.

OBJECTIVES OF OPERATING SYSTEM

Operating system has three main objectives

- **Convenience**: An operating system makes a computer system convenient and easy to use, for the user.
- **Efficiency:** An operating system allows the computer to use the computer hardware in an efficient way, by handling the details of the operations of the hardware.
- **Ability to Evolve:** An operating system should be constructed in such a way as to permit the effective development, testing, and introduction of new system functions without at the same time interfering with service.

NEEDS AND SERVICES OF OPERATING SYSTEM/ FUNCTIONS OF OPERATING SYSTEM

Operating System performs a number of functions for the computer system that are follows:

1) ***It acts as an Command Interpreter:***
   - Generally the CPU cannot understand the commands given by user. It is thefunction of operating System to translate this command (human understandable) into m/c understandable instructions that the system (CPU) can understand.
   - After the execution of instructions by CPU, it retranslates the o/p back into a human understandable language.
   - To execute the user jobs, the Operarting System interacts with the computer hardware.
2) ***It acts as the Resource Manager:***
   - An Operating System acts as a resource manager in two ways
     - Time multiplexing
     - Space multiplexing
   - In time multiplexing the different resources (hardware or software) can beshared among different users for a optimal or fixed time slot.
   - e.g. the Operating System allocates a resources such as CPU to program A for a fixed time slot. When the time slot of process A is over, the CPU is allocated to another program
     B. If program A needs more CPU attention, then the CPU again allocated to program A after the time slice period allocated to program B is over.
   - In space multiplexing, different resources are shared at the same time among different programs .e.g. sharing of hard disk and main memory by different users at the same time.
3) ***Memory Management:***
   - It keeps track of the resources (memory),what part of memory is in use and by whom, which part of the memory is not in use.
   - Decides which processes are to be loaded when memory space is available.
   - Allocation and de allocation of memory
4) ***Process Management:***
   - A process(task) is an instance of a program in execution. A program is just a passive entity, but a process is an active entity.
   - To accomplish its task ,a process needs certain resources like CPU time, memory,files and I/O devices.
   - These resources are allocated to process either at the time of creation or when it is executing.

- The operating system is responsible for the following functions related to process management.
    i. Process creation (loading the prog. From secondary storage to main memory)
    ii. Process scheduling
    iii. Provide mechanism for process synchronization
    iv. Provide mechanism for deadlock handling
    v. Process termination

5) ***Peripheral or I/O device Management:***
    - Keep track of resources (device, channels, control units) attached to the system.
    - Communication between these devices and CPU is observed by operating system.
    - An operating system will have device drivers to facilitate I/O functions involving device likekeyboard, mouse, monitor, disk, FDD, CD-ROM, printer etc.
    - Allocation and De allocation of resources to initiate I/O operation.
    - Other management activities are
        i. Spooling
        ii. Caching
        iii. Buffering
        iv. Device driver interface

6) ***File Management:***
    - A file is a collection of related information or record defined by the user.
    - The operating system is responsible for various activities of file management are
        i. Creation and deletion of files
        ii. Creation and deletion of directories
        iii. Manipulation of files and directories
        iv. Mapping files onto secondary storage

7) ***Secondary storage Management:***
    - It is a larger memory used to store huge amount of data. Its capacity is much larger than primary memory. E.g. floppy disk, hard disk etc.
    - The operating system is responsible for handling all the devices that can be done by thesecondary storage management.
    - The various activities are:
        i. Free space management
        ii. Storage allocation (allocation of storage space when new files have to bewritten).
        iii. Disk scheduling (scheduling the request for memory access)

8) ***Protection/Security Management:***
    - If a computer system has a multiple processor, then the various processes must beprotected of one another's activities.
    - Protection refers to mechanism for controlling user access of programs or processes or user to resources defined by the computer system.

9) ***Error detection and Recovery:***
    - Error may occur during execution like divide by zero by a process, memory access violation, deadlock, I/O device error or a connection failure.
    - The operating system should detect such errors and handles them.

CLASSIFICATION / TYPES OF OPERATING SYSTEM

All operating System consists of similar component and can perform all most similar function but the methodand procedure for performing these functions are different.

OPERATING SYSTEM are classified into different categories according to their different features. Thefollowing section will discuss the classification of operating system.

Single user OPERATING SYSTEM:

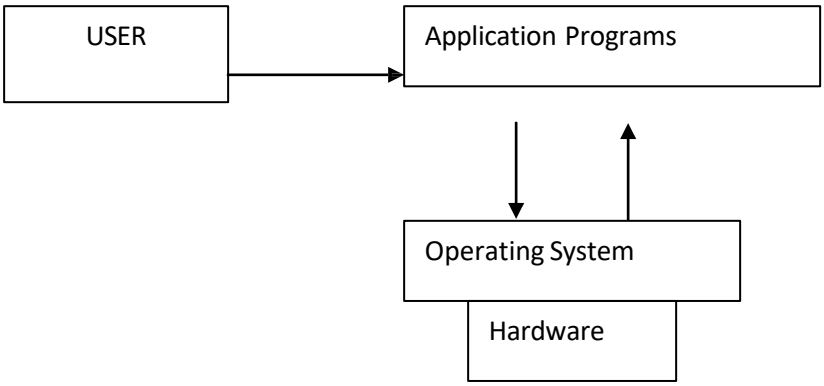
- In a single user operating system a single user can access the computer at a particular time.
- This system provides all the resources to the user at all the time.
- The single user operating System is divided into the following types.
  - Single user, single tasking operating System
  - Single user , multitasking operating System

Single user, single tasking operating System:

- In a single user, single tasking operating system, There is a single user to execute a program at aparticular system.
- Example – MS-DOS

Single user , multitasking operating System

- In single user, multitasking OPERATING SYSTEM a single user can execute multiple programs.
- Example – A user can program different programs such as making calculations in excel sheet, printing a word document & downloading into the file from internet at the sametime.



Advantage:

- The CPU has to handle only one application program at a time so that process management is easy in this environment.

Disadvantage:

- As the operating system is handling one application at a time most of the CPU time is wasted.

Multi user OPERATING SYSTEM:

- In a multi-user operating system , multiple number of users can access different resources of a computerat a time.
- This system provides access with the help of a network. Network generally consists of various personal computers that can and receive information to multi user mainframe computer system.
- Hence, the mainframe computer acts as the server and other personal computer act as the client for that server.
- Ex: UNIX, Window 2000

Advantage:

Sharing of data and information among different user.

Disadvantage:

Use of expensive hardware for the mainframe computer.

Batch Operating System

- In a batch processing operating system interaction between the user and processor is limited or there is no interaction at all during the execution of work.
- Data and programs that need to be processed are bundled and collected as a 'batch'.
- These jobs are submitted to the computer through the punched card. then the job with similar needs executed simultaneously.

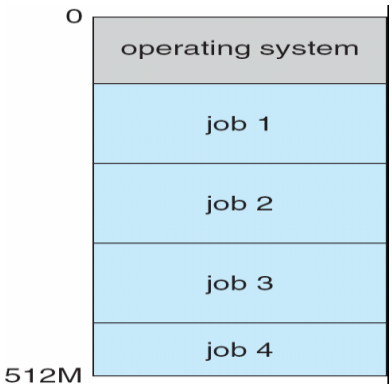
Advantage:

It is simple to implement.

Disadvantage:

Lack of interaction between user and the program.

Multiprogramming OPERATING SYSTEM:



- In a multiprogramming operating System several user can executemultiple jobs by using a single CPU at the same time.
- The operating System keeps several program or job in the mainmemory.
- When a job is submitted to the system in a magnetic disk or job pool.
- Then some of the jobs are transferred to the main memory according to the size of the main memory.
- The CPU execute only one job which is selected by theoperating System.
- When the job requires any I/O operation, then CPU switches to next job in the main memory i.e CPU do not have to wait for the completion ofI/O operation of that job.
- When the I/O operation of that job is completed then the CPU switches to the next jobafter the the execution of the current job.
- E,g.UNIX, Windows 95 etc

Advantage:

CPU utilization is more i.e the most of the time the CPU is busy.

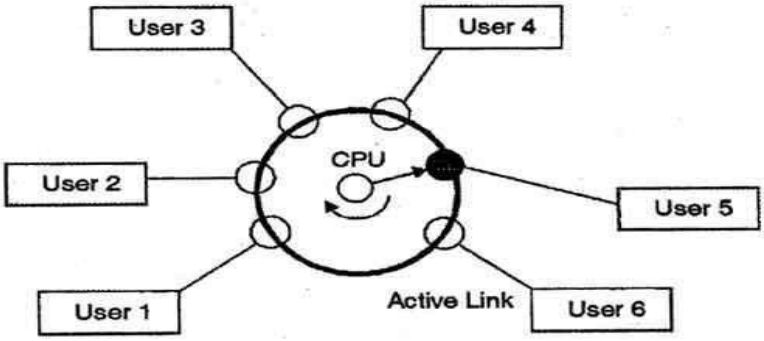
Disadvantage:

The user can't directly interact with the system.

Time sharing Operating System:

- This is the Logical extension of multiprogramming system.
- The CPU is multiplexed among several jobs that are kept in memory and on disk (the CPU is allocated to a job only if the job is in memory).

- Here the CPU can execute more than one job simultaneously by switching among themselves.
- The switching process is very fast so that the user can directly interact with the system during the execution of the program.
- This system stores multiple jobs in the main memory and CPU execute all the jobs in asequence.
- Generally CPU time is divided into no. of small interval known as **time slice period**.
- Every process has to execute for the time slice period; then the CPU switch over to nextprocess.
- The switching process is very fast,so it seems that several processes are executed simultaneously.



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.

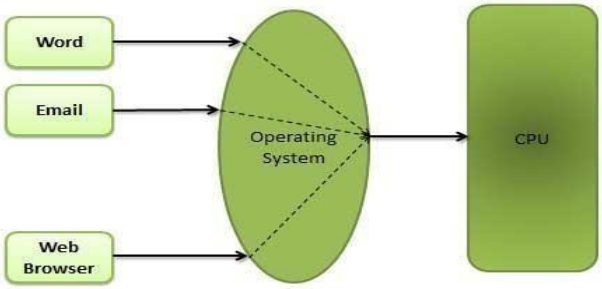
Advantage:

- CPU utilization is more i.e the most of the time the CPU is busy.

Disadvantage:

- The operating system is more complex due to memory management, Disk management etc.

Multitasking Operating System:



- A multi-tasking operating system allows more than one program to be running at the same time.
- E.g.-one user can open the word document and can simultaneously access the internet.
- While the processor handles only one application at a particular time it is capable of switching between the applications effectively to apparently simultaneously execute each application.
- This type of operating system is seen everywhere today and is the most common type of operating system, the Windows operating system would be an example.

Multiprocessing Operating System:

- When a system contains more than one processor in close communication, sharing the computer bus, the clock and sometimes memory and peripheral devices is known as **multiprocessing operating System.**

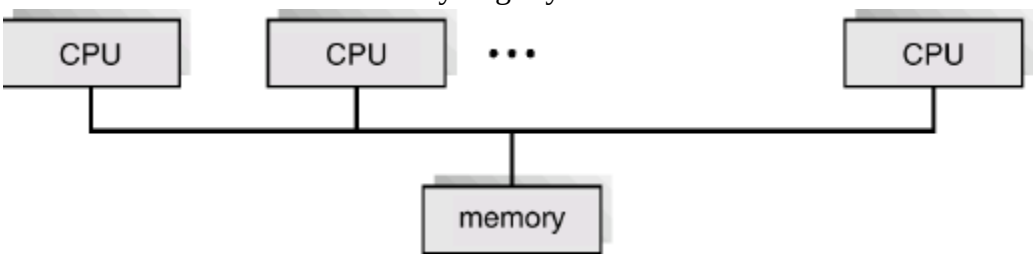
- This is divided into 2 types:
- Symmetric multiprocessing system
- Asymmetric multiprocessing system

Symmetric multiprocessing (SMP)

- Each processor runs a shared copy of the operating system.
- Different processor can communicate with each other and are able to execute this copy at the same time.
- These processor are executed by a single operating System and have equal right to access all theI/O devices connected to the system.

Asymmetric multiprocessing(ASMP)

- It is based upon the principle of master slave relationship.
- In this system one processor runs the operating System and other processor run the userprocesses.
- The processor which runs the operating System is known as master processor the processorwhich runs the user processes known as the slave processor.
- It is used in large system.
- Each processor is assigned a specific task; master processor schedules and allocated work to slave processors.
- More common in extremely large systems



Advantage:

- **Improved Reliability**:-As the system consists of multiple processor, failure of one processor does not disturb the computer system. The other processor in the system continues the task.
- **Improved throughput**:-throughput is defined as the total no of jobs which are executed by the CPU in one second. As this system use multiple processor all the workload divided between the different processor.
- **Economica**l:-in this system different processor share the clock, bus, peripheral and memory between them. Due to this reason the system are more economical than multiple single processor system.

Real time Operating System:

- In a real-time operating system a job is to be completed within the right time constraint otherwise job loses its meaning.
- These system compete a particular job in the fixed time slot in order to respond to an event quickly.
- Real time introduces for correct operation and it required to produce result within a non negotiable time period.
- Real-time systems are usually used to control complex systems that require a lot of processing like machinery and industrial systems.

- This is of 2 types:
  - Hard real time operating system
  - Soft real time operating system

Hard real-time system:

- This system completes the critical tasks within the definite interval of time constraint.
- If the critical task is not completed within the time constraint, then the system fails.
- This system has to complete all the processes within the definite deadline and a single miss leads to critical failure.
- E.g.:-Pace maker, flight control system(any miss in deadline leads to crash).

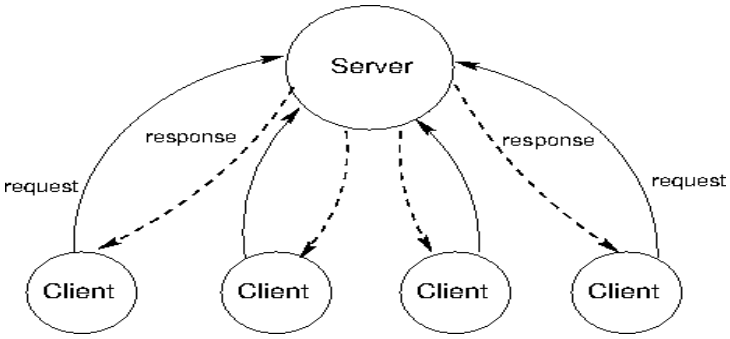
Soft real-time system:

- These systems are not affected by the lapse of time interval and do not cause any critical failure.
- E.g.:-Live video streaming.

Distributed Operating Systems:

- In distributed operating system , the users access remote resources in the same way as the local resources are accessed.
- Distribute the computation among several physical processors.
- 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.
- These systems provide features such as data and process migration.
- This operating system based on two models.
  - Client-server model
  - Peer –to-peer model

Client-server model:-In this model, the client sends a request for a resource to the server and the server, in turn provides the requested resource as a response back to the client.



Peer –to-peer model: In a peer-to-peer model,all the computers behave
as peers as well as clients. These peers communicate with each other for exchange of their resources.

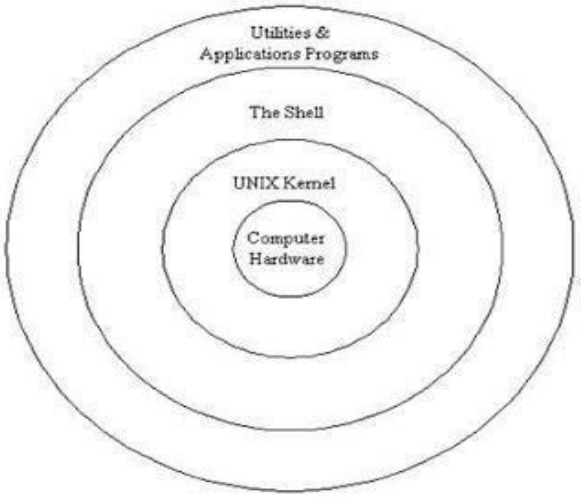
***Advantages**:*

- It facilitates the sharing of hardware and software resources between different processors.
- It increases reliability as failure of one node does not affect the entire network.
- It increases the computational speed of computer by sharing the workload into different nodes.
- It enable different users to communicate with each other using email.

Structure of Operating System

The structure of Operating System comprises of 4 layers.

- Hardware
- Kernel
- System call interface(shell)
- Application programs



Kernel:

- It is the vital part of the operating system. It interacts directlywith the hardware of a computer.
- Programs interact with the kernel through 10 system calls.
- **System call:-** The *system call* provides an interface to the operating system services.
- System call tells the kernel to carry out various asks for the program such as opening a file, writing to a file, obtaining information about a file, executing a program, terminating a process etc.
- The main function of the **kernel** are
  - To manage computer memory
  - To maintain file system
  - Allocation of resources
  - Control access to the computer
  - Handle interrupts

System Call Interface(Shell):

- Shell is command line interpreter which interprets the commands given by the user.
- It is software or program which acts as a mediator between kernel and user.

- The shell read the commands, what you typed at command line and interprets them and sends request to execute a program. That's why shell is called as command line interpreter.

Hardware:

- Computer hardware refers to the physical parts or components of a computer such as the monitor, mouse, keyboard, computer data storage, hard drive disk (HDD), system unit (graphic cards, sound cards, memory, motherboard and chips), etc. all of which are physical objects that can be touched

Utility and application programs:

- Utility programs help **manage, maintain and control computer resources**. These programs are available to help you with the day-to-day chores associated with personal computing and to keep your system running at **peak performance**.
- Application software is all the computer software that causes a computer to perform useful tasks beyond the running of the computer itself.
- Examples of application programs include word processors; database programs; Web browsers; development tools; drawing, paint, and image editing programs; and communication programs.

Evolution of Operating system:

1. Serial operating system.
2. Batch operating system.
3. Multiprogramming operating system.
4. Time-Sharing operating system.
5. Real-Time operating system.
6. Multiprocessing operating system
7. Distributed operating system

UNIT-2

PROCESS MANAGEMENT

PROCESS:

- Process is a program in execution
- Process is a currently executable task.
- Process execution must progress in a sequential manner.

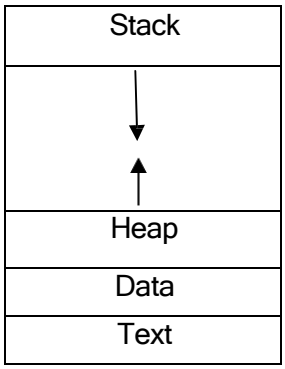
Process	Program
i) A process is the set of executable instruction, those are the machine code.	i) It is a set of instruction written in programming language.
ii) Process is dynamic in nature.	ii) Program is static in nature.
iii) Process is an active entity.	iii) Program is a passive entity.
iv) A process resides in main memory.	iv) A program resides in secondary storage.
v) A process is expressed in assembly language or machine level language.	v) A program is expressed through a programmable language.
vi) The time period limited.	vi) Span time period is unlimited.

Process in Memory:-

⇒ A process resides in memory through following section i.e.

1) Stack
2) Heap
3) Data
4) Text

- Stack section contains local variable
- Data section contains global variable
- Text section contains code or instruction.
- Heap section contains memory which will be dynamicallyallocated during runtime.



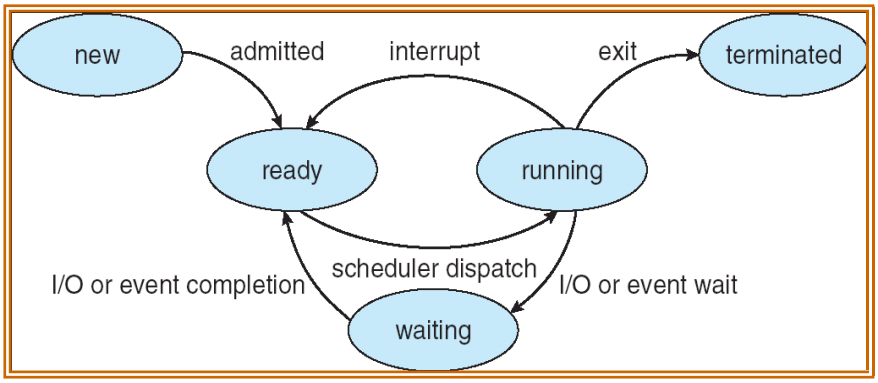
PROCESS STATE:

When a process is executed, it changes its state. The current activity of that process is known asProcess state.A process has different states. They may be

➢ **New state**:

- When the request is made by the user, the process is created.

- The newly created process moves into a new statement.
- The process resides in secondary memory through a queue named as job queue or job pool.



⇒.

Diagram of process state

- **Ready state:-**
  - A process is said to be ready if it needs the CPU to execute.
  - Out of total newly created processes, specified processes are selected and copied to temporary memory or main memory.
  - In main memory they resides in a queue named as ready queue.
- **Running:-**

A process is said to be running if it moves from ready queue and starts execution using CPU.

- **Waiting state/ blockedstate:-**
  - A process may move in to the waiting state due to the following reasons.
    - If a process needs an event to occur or an input or output device and the operating system does not provide I/O device or event immediately, then the process moved into a waiting state.
    - If a higher priority process arrives at the CPU during the execution of an ongoing process, then the processor switches to the new process and current process enter into the waiting state.
- **Terminated state:-**
  - After completion of execution the process moves into the terminated state by exiting the system. The terminated state converts the process into a program.
  - Sometimes operating system terminates the process due to the following reasons.
    - Exceeding the time limit
    - Input/output failure
    - Unavailability of memory
    - Protection error
    -

PROCESS CONTROL BLOCK(PCB)/ TASK CONTROL BLOCK(TCB)

- To represent a process the operating System needs to group all the information of a process inside a datastructure. This data structure is known as **process control block(PCB).**
- In other words operating System represents each process by a PCB. An operating System considers a process as afundamental unit for Resource Allocation. Following resources could be allocated to a process.

The information stored inside the PCB includes

i. **Pointer-**It stores the starting address of the process.
ii. **Process State-** This field stores or represent the current state of the process whether it s in ready/running/new/waiting/terminating.
iii. **Process ID/Number-**Each process has uniqueID or serial no. Each process is shown an unique no. known as its Process ID or ProcessNumber.
iv. **Program Counter-** It stores the address of the next instruction to be executed.
v. **Register-**This field contains the name of the registers which are currently used by the processor.
vi. **Scheduling Information-**This field stores the information about the scheduling algo.used by operating System for scheduling that process.
vii. **Memory Management Information-**This field contains the value of thebase table, segment table and page table.
viii. **Account Information-**This field contains the total no. processes, time slice period it used.
ix. **File Management Information-** It stores various information about the files used by the process.
x. **I/O Status Information-**It stores the information about various allocated I/O devices to the process, a list of open files & so on.

PROCESS SCHEDULING

- The objective of multiprogramming is to have some process running at all times, to maximize CPU utilization.
- The objective of time sharing is to switch the CPU among processes so frequently that users can interact with each program while it is running.
- This purpose can be achieved by keeping the CPU busy at all the times.
- So, when two or more processes compete for the CPU at the same time, a choice has to be made.
- This procedure of determining the next process to be executed on the CPU is called as **Process Scheduling.**
- The module of the operating system that makes this decision is called as **Schedule**r.
- Process scheduling consists of three sub functions:

I. Scheduling Queue
II. Scheduler
III. Context Switching

I. Scheduling Queue

The operating system maintains several queues for efficient management of processes. These are as follows:

1. Job Queue:

- When the process enters into the system, they are put into a job queue.
- This queue consists of all processes in the system on a mass storage device such as hard disk.

2.Ready Queue:

- From the job queue, the processes which are ready for execution are shifted to the main memory.
- In the main memory the processes are kept in the **ready queue**.
- In other words, the ready queue contains all those processes that are waiting for the CPU.

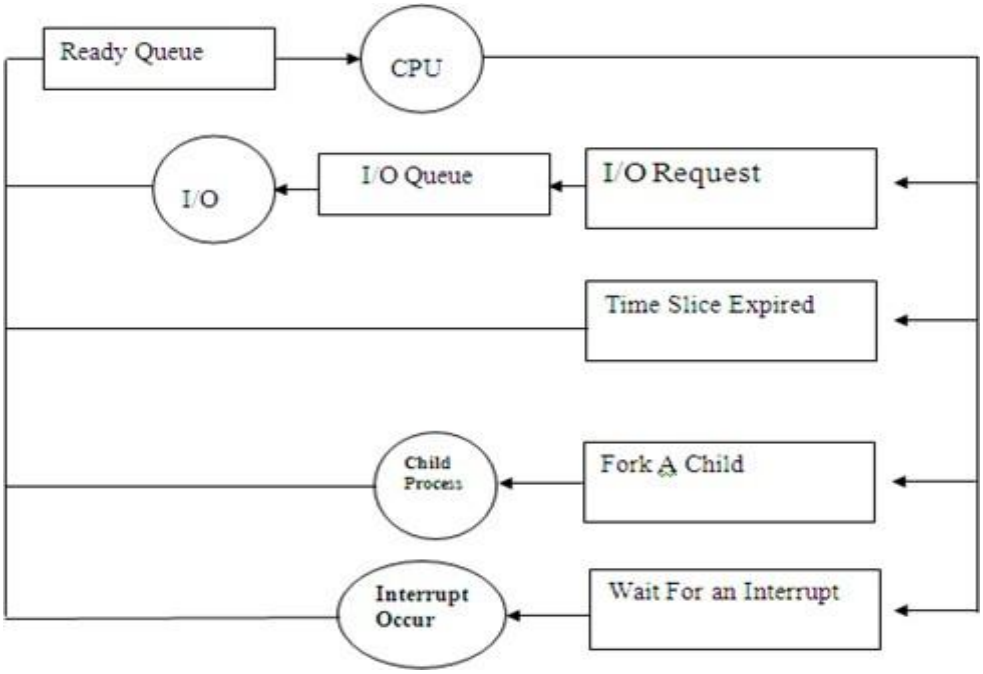
3.Device Queue:

- Device queue is a queue for which a list of processes waiting for a particular I/O device.
- Each device has its own device queue.
- When a process required some I/O operation, it is then taken out of the ready queue and kept under the device queue.

4.**Suspended Queue:** It stores the list of suspended process.

Queuing Diagram:

- The process could issue an I/O request and then be placed in an I/O queue.
- The process could create a new subprocess and wait for its termination.
- The process could be removed forcibly from the CPU as a result of an interrupt, and again put back in the ready queue.



II. Scheduler:

- The module of the operating system that makes the decision of process scheduling is known as **Scheduler**.
- Their main task is to select the jobs to be submitted into the system and to decide which process to run.

- Schedulers are of three types.

  1. Long Term Scheduler
  2. Short Term Scheduler
  3. Medium Term Scheduler

Long Term Scheduler(LTS)

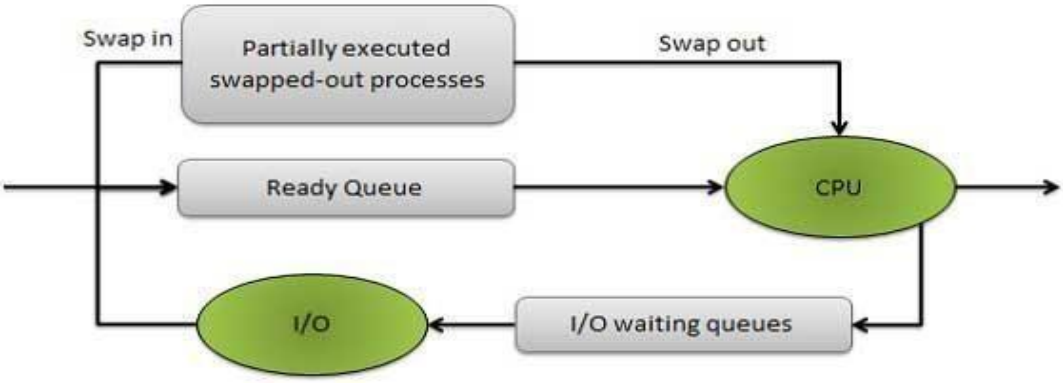
- It is also called **job scheduler**; it works with the job queue.
- Job scheduler selects processes from the job queue and loads them into the main memory for execution.
- It executes much less frequently, as there may be long time gap between the creation of new process in the system.
- The **primary objective** of the job scheduler is to control the degree of multiprogramming.
- If the degree of multiprogramming is stable, then the average rate of process creation must be equal to the average departure rate of processes leaving the system.
- When process changes the state from new to ready, then there is a long term scheduler.
- LTS selects a balanced mix of CPU bound and I/O bound processes.

Short Term Scheduler(STS):

- It is also called **CPU scheduler** or **process scheduler**.
- It selects the process from ready queue and allocates CPU to it.
- Main objective is to increase the system performance.
- This scheduler is frequently invoked as compared to Long term scheduler.
- It is the change of ready state to running state of the process.
- This is faster one because the process executes for short time period before waiting for an I/O request.

Medium Term Scheduler(MTS):

- It is also known as Swapper.
- Sometimes the processes are removed from the memory and from CPU to reduce the degree of multiprogramming.
- Then after sometime the processes can be reintroduced into memory and execution can be continued where it is left off. This scheme is known as swapping.
- The Medium Term Scheduler selects a process among the partially executed or unexecuted swapped out processes and swaps it in the main memory.



III. Context Switching

- Transferring the control of the CPU from one process to other requires saving the context of currently running process and loading the context of another ready process. This mechanism of saving and restoring the context is known as **context switch**.
- The portion of the PCB including the process state, memory management information and CPU scheduling information together constitutes the **Context** or **State** of the process.
- The switching periods depends upon the memory speed and the number of registers used.

CPU SCHEDULING

Basic Concept:

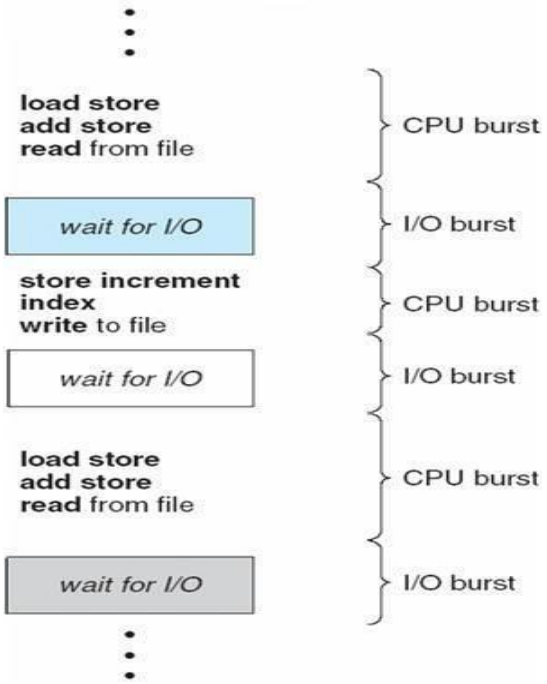
The objective of multiprogramming is to improve the productivity of the computer. It can be done by maximizing the CPU utilization. That means some process running at all times and this is happened by switching the CPU among processor.

But in a unipolar system only one process may run at a time and other processes must wait until the CPU is free and can be rescheduled.

Scheduling is a fundamental operating system function. Almost all computer resources are scheduled before use. The CPU is one of the primary computer resource. Thus its scheduling is to control the operating System design.

CPU-I/O Burst Cycle

- The success of CPU scheduling depends on an observed property of processes.
- Process execution consists of a cycle of CPU execution and I/O wait.
- Processes alternate between these two states. Process execution begins with a CPU burst.
- That is followed by an I/O burst, which is followed by another CPU burst, then another I/O burst, and so on.
- The final CPU burst ends with a system request to terminate execution.



CPU Scheduler(Short Term Schedular):

- Whenever the CPU becomes idle, the operating system must select one of the processes in the ready queue to be executed.
- The selection process is carried out by the short-term scheduler (or CPU scheduler).
- Scheduling can be of 2 types:
  - Non-Preemptive Scheduling
  - Preemptive Scheduling

Non-Preemptive Scheduling:
In this case once the CPU has been allocated to a process, the process keeps the CPU until it releases the CPU by terminating or by switching to the waiting state. That is when it is (process)is computed or required any I/O operation.

Preemptive Scheduling:
In this case CPU can be released forcefully**.** Under this scheduling the process has to leave the CPU forcefully on the basis of criteria like running to ready an d waiting state to ready state(i.e. when interrupt occur or due to completion of time slice period).

DISPATCHER:

- Dispatcher is the module that gives control of the CPU to the process selected by the short-term scheduler.
- The time it takes for the dispatcher to stop one process and start another running is known as the **dispatch latency.**
- This function involves the following:
  - Switching context
  - Switching to user mode
  - Jumping to the proper location in the user program to restart that program.

Scheduling Criteria:

- There are several CPU scheduling algorithm. But we have to select one which is suitable for our system.

- There are some criteria based on which CPU scheduling algorithm select the next process to execute.

  - **CPU utilization**. We want to keep the CPU as busy as possible. Conceptually, CPU utilization can range from 0 to 100 percent. In a real system, it should range from 40 percent (for a lightly loaded system) to 90 percent (for a heavily used system).

  - **Throughput:** it can be defined for a system as "the no. of jobs completed per unit time".

  - **Turnaround time:** The interval of time from submission of a process to the time of completion. It is the total time spent by a process within the system.
    Turnaround time= time spent in the ready queue + time spent in execution + time spentin I/O operation.

    OR

    Turn Around time = Completion time – Arrival time

- **Waiting time**: It is the sum of the periods spent waiting in the ready queue .[that means CPU scheduling affects only the amount of time that a process spends waiting in the ready queue , but does not affect the amount of time during which process executes or does I/O].

  Waiting time = Turn Around time – Burst time

  It is the amount of time during which process is in the ready queue.
- **Response time:** It is the amount of time , process takes to start responding(first response after submission)

  Response Time = Time at which process first gets the CPU – Arrival time

SCHEDULING ALGORITHM

The Scheduling algorithm decides which of the process in ready queue is to be attending the CPU.There are various scheduling algorithms:

1. First Come First Serve scheduling(FCFS)
2. Shortest Job First(SJF)
3. Priority scheduling
4. Round Robin Scheduling
5. Multilevel Queue scheduling

First Come First Serve scheduling(FCFS)

- This is the simplest and easiest scheduling algorithm.
- In this scheme, the process that requests the CPU first is allocated the CPU first.
- The first process is stored in the first position of the ready queue.
- Here the data structure of the ready queue is FIFO queue.
- FCFS is non preemptive. when CPU is free, it is allocated to other process i.e the CPU has been allocated to process, that process keeps the CPU until it release the CPU either by terminating or by requesting I/O.

- Let the process arrives in the order p1, p2, P3, p4,p5.

Process	Arrival Time	CPU Burst
P1	**0**	**20**
P2	**4**	**2**
P3	**6**	**40**
P4	**8**	**8**
P5	**10**	**4**

Find out the Average Turn Around Time(ATAT) and Avg. Waiting Time(AWT).

Solution:

The result of execution shown in GANTT CHART:

P1	P2	P3	P4	P5

0 20 22 62 70 74

Waiting time:

P1=0
P2=20-4=16
P3=22-6=16
P4=62-8=54
P5=70-
10=60

Hence the AWT(Average Waiting Time)=(0+16+16+54+60)/5=29.2

Turn Around Time(TAT):

P1=20-0=20
P2=22-4=18
P3=62-6=56
P4=70-8=62
P5=74-
10=64

Hence **Average TAT=(20+18+56+62+64)/5=44**

Disadvantage:

- The user having small job has to wait for a long time.
- This algorithm is particularly troublesome for tie sharing system because each user needs to geta share of the CPU at regular time intervals.

Advantage:

- FCFS scheduling is very simple to implement and understand.

<u>Shortest Job First Scheduling(SJF)</u>

- In this type of scheduling when the CPU is available , it is assigned to the process that has the smallest next CPU burst.
- If two processes have the same length next CPU burst, FCFS scheduling is used to break the tie.
- It is also known as shortest next CPU burst.
- SJF algorithm may be either preemptive or non preemptive.
  - The choice arises when a new process arrives at the ready queue while a previous process is executing.
  - The new process may have a shortest next CPU burst than the currently executingprocess.
  - A preemptive SJF algorithm will pre-empt the currently executing process where as a non premptive SJF algorithm will allow the currently running process to finish its CPU burst.

- Preemptive SJF scheduling is sometimes called "shortest remaining time first scheduling".
- Larger jobs will never be executed if smallest jobs arrives.

Process	Arrival Time	CPU Burst
P1	00	15
P2	05	10
P3	07	05
P4	10	08

Non preemptive

Gantt Chart:

P1	P3	P4	P2

0 15 20 28 38

Waiting Time:

P1=0

P3=15-7=8

P4=20-10=10

P2=28-5=23

AWT=0+8+10+23=41/4=10.25

Turn Around Time:

P1=15

P2=38-5=33

P3=20-7=13

P4=28-10=18

ATAT=(15+33+13+18)/4=19.75

Priority scheduling:

In case of priority scheduling the process having highest priority value will be executed first.

Problem:

processs	AT	BT	Priority
P1	00	15	3
P2	04	10	2
P3	06	05	1
P4	08	08	4

SOLUTION(NON-PREEMPTIVE GANTT CHART):

P1	P3	P2	P4

0 15 20 30 38

W.T.

P1=0

P2=20- 4=16

P3=15-6=9

P4=30-8=22
A.W.T=0+16+9+22=47/4=11.75

T.A.T

P1=15- 0=15
P2=30-4=26
P3=20-6=14
P4=38-8=30
A.T.A.T=15+26+14+30=85/4=21.25

Internal Priority:

In Priority Scheduling a priority value is assigned to each of the process in the ready queue. The priority value can be assigned either internally or externally. The factor for assigning internal priority is:

- Burst time
- Memory Requirement
- I/O devices
- No. Of files

External Priority:

The factor for assigning the external priority value are:

- Important of process
- Amount of fund given
- Political pressure

The priority scheduling may be preemptive or non-preemptive.the major problem with the priority scheduling is indefinite blocking or starvation. The solution to the problem is **aging**. Aging is a technique which gradually increases the priority value of the process that waits in the ready queue for a long time.

Problem:

Process	**B.T**	**Priority**
P1	10	3
P2	1	1
P3	2	4
P4	1	2

Gantt Chart:

P2	P4	P1	P3

0 1 2 12 14

W.T

P1=2 P2=0 P3=12 P4=1

A.W.T= (2+0+12+1)/4=3.75

T.A.T

P1=12 P2=1 P3=14 P4=2

A.T.A.T=(12+1+14+2)/4=7.25

Round Robin Scheduling:

- This is designed for Time sharing system. It is similar to FCFS scheduling.

- But the CPU pre-empts among the ready process in every time slice period, which are in the ready queue.
- In case of FCFS scheduling the ready queue is a FIFO queue. But in RR scheduling the readyqueue is a circular queue.
- Round Robin Scheduling is a purely preemptive scheduling algorithm. Because after every time slice period CPU will switch over to the next process in the ready queue.

Process	A.T.	B.T.
P1	00	20
P2	10	10
P3	15	15
P4	15	10

CPU Time=5ms

P1	P1	P2	P3	P4	P1	P2	P3	P4	P1	P3

0 5 10 15 20 25 30 35 40 45 50 55

W.T:

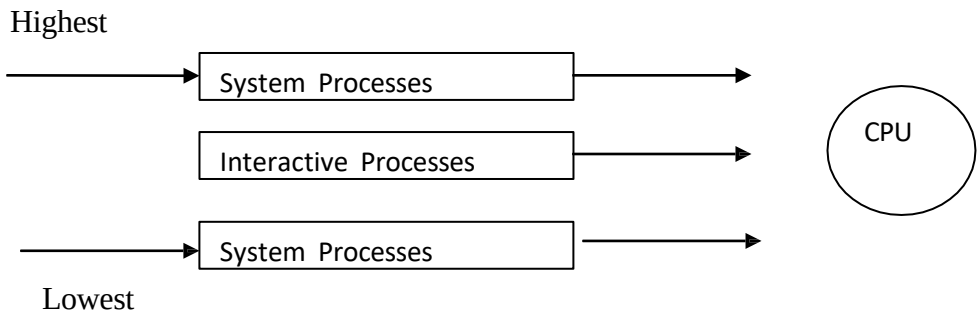
P1=(25-10)+(45-30)=30
P2=30-15=15
P3=(35-20)+(50-40)=20
P4=(20-15)+(40-25)=20
A.W.T=(30+15+20+20)/4=21.25

T.A.T

P1=50
P2=(35-10)=25
P3=(55-15)=40
P4=(45-15)=30
A.T.A.T=(50+25+40+30)/4=36.25

Multilevel Queue scheduling

- This algorithm partitions the ready queue into several separate queues.
- Processes are permanently assigned to one queue based on some criteria such as memory size, process priority.
- Each queue has its own scheduling algorithm.
- Foreground queue may be in RR and background queue may be in FCFS.
- Again there is a scheduling algorithm to select a queue among many queues.
- If priority queue is applied .then no process in the lowest priority queue can be executed till there is a process in the higher priority queue.
- If Round Robin scheduling is applied, then each queue gets CPU for a certain amount of time. Again that time will be divided among the processes in that queue.



Interprocess Communication(IPC)

Overview :

Processes are classified into 2 categories.

They are:i)Independent process

ii)Cooperating process

Independent process:-

It is defined as a process that does not share any data and does not communicate with other process.
In other words we can say that modification made to an independent process does not affect the functioning of other processes.

Co-operating process:-

It is defined as a process, which gets affected by any other process.

These processes are used for resource sharing and to speed up a computation procedure.

Interprocess Communication(IPC)

Interprocess communication is the mechanism provided by the operating system that allows processes to communicate with each other.

Processes are classified into 2 categories. They are:

Independent process: An independent process is not affected by the execution of other processes.

Cooperating process: a co-operating process can be affected by other executing processes.

Advantages of process cooperation

Information sharing: Since several users may be interested in the same piece of information (for instance, a shared file), we must provide an environment to allow concurrent access to these types of resources.

Computation speedup: If we want a particular task to run faster, we must break it into subtasks, each of which will be executing in parallel with the others. Such a speedup can be achieved only if the computer has multiple processing elements (such as CPUS or I/O channels).

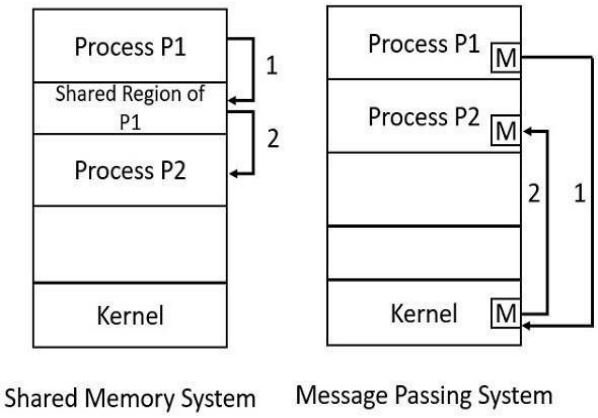
Modularity: We may want to construct the system in a modular fashion, dividing the system functions into separate processes or threads.

Convenience: Even an individual user may have many tasks on which to work at one time. For instance, a user may be editing, printing, and compiling in parallel.

Ways to Implement IPC

1. **Shared Memory:** Multiple processes can access a common shared memory. Multiple processes communicate by shared memory, where one process makes changes at a time and then others view the change. Shared memory does not use kernel.

2. **Message Passing:** Message passing provides a mechanism to allow processes to communicate and to synchronize their actions without sharing the same address space. It is very useful in case where the tasks or processes reside on different computers and are connected by a network. Messages can be of fixed or variable size.



UNIT-3

MEMORY MANAGEMENT

One of the major functions of operating system is memory management. It controls the

- Allocation and de-allocation of physical memory.
- Which part of the memory is currently used by which process.
- Decide which processes are to be loaded into memory.
- Free space management.
- Dynamic allocation/de-allocation of memory to executing processes etc.

Logical Address & Physical Address:-

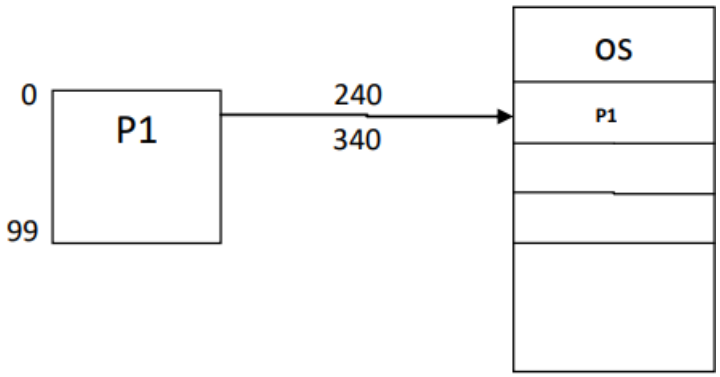
Logical Address: Address generated by a CPU is called as logical address.

Physical Address:-Address generated by memory management unit is called as physical address.The logical address is known as "virtual address".

The set of all logical unit or address generated by programs referred as "logical address space".

The set of physical address corresponding to logical address is referred as "physical address space".Suppose the program size= 100 KB

But it is loaded in the main memory from 240 to 340 KB.

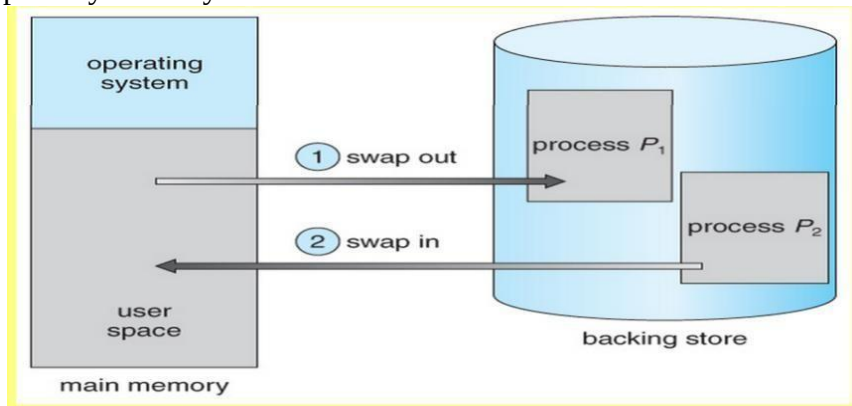


- So 0 to 99 KB is the logical address space but 240 to 340 KB is the physical address space.
- Physical address space = logical address space + content of relocation register.
- The mapping between logical and physical addresses are done at run-time by the memorymanagement unit (MMU).

SWAPPING:-

- Swapping is the method to improve main memory utilization.
- When a process is executed it must be in the main memory.
- A process can be swapped out temporarily to secondary memory or hard disk or backing memory and then again brought back to secondary memory for execution. This technique is known as "**Swapping".**
- The basic operation of swapping is
    - Swap-out (roll-out)
    - Swap-in (roll-in)

Swap-out:-The mechanism to transfer the process from main memory to secondary memory. Swap-in:- The mechanism that shifts the process from secondary memory to primary memory.



MEMORY ALLOCATION METHODS

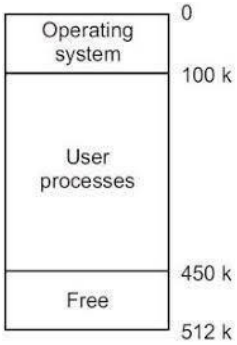
- The main memory must accommodate both operating system and various user processes.
- Generally, the main memory is divided into 2 partitions.
  - Operating system.
  - Application program/user processes.
- The operating system place in either low memory or high memory.
- Commonly the operating system is loaded in low memory.
- Generally, there are two methods used for partitioning the memory allocation.
  - Contiguous memory allocation
  - Non-Contiguous memory allocation

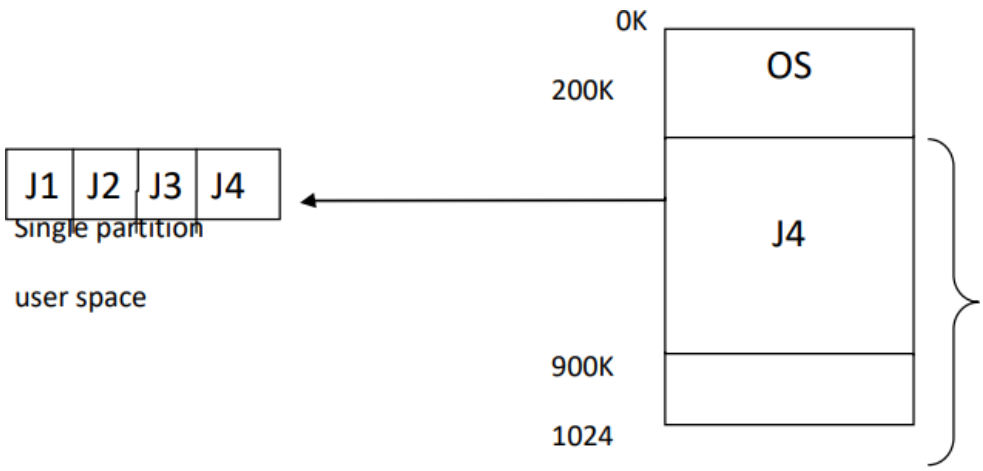
Contiguous Memory Allocation:-

- It is again divided into two parts.
  - Single partition allocation.
  - Multiple partition allocation.

Single Partition Allocation:-

- In this memory allocation method, the operating system reside in the low memory.





- And the remaining part/space will be treated as a single partition.
- This single partition is available for user space/application program.
- Only one job can be loaded in this user space is the main memory consisting of only one processat a time, because the user space treated as a single partition.

Advantage:-

i. It is very simple.
ii. It does not require expertise to understand.

Disadvantage:-

i. Memory is not utilized property.
ii. Poor utilization of processor (waiting for I/O).

Multiple Partition Allocation:-

This method can be implemented in 3 ways. These are:

- Fixed equal multiple partition.
- Fixed variable multiple partition.
- Dynamic multiple partition.

Fixed equal multiple partition:-

i. In this memory management scheme the operating system occupies the low memory and rest of mainmemory is available for user space.
ii. The user space is divided into fixed partitions. The partition size depends upon the operating system.
iii. A partition of main memory is wasted within a partition is said to be “Internal Fragmentation”and the wastage of an entire partition is said to be ”External Fragmentation”.
iv. There is one problem with this method is memory utilization is not efficient which causesinternal and external fragmentation.

Advantages:-

- This scheme supports multiprogramming.
- Efficient utilization of CPU & I/O devices.
- Simple and easy to implement.

Disadvantages:-

- This scheme suffers from internal as well as external fragmentation.
- Since, the size of partitions are fixed, the degree of multiprogramming is also fixed.

<u>Fixed variable partition:- (unequal size partition)</u>

- In this scheme the user space of main memory is divided into number of partitions, but thepartitions sizes are different length.
- The operating system keep a table which indicates, which partition of memory are available and which areoccurred. This table is known as "Partition Description Table" (PDT).
- When a process arrives and needs allocation or memory, we search for partition which is bigenough to allocate this process. If find one allocation, then allocate the partition to that process.

Advantage:-

i. Supports multiprogramming.
ii. Smaller memory loss (expected).
iii. Simple & easy to implement.Disadvantage:-

i. Suffers from internal as well as external fragmentation.

Figure 4: Fixed Partitioning

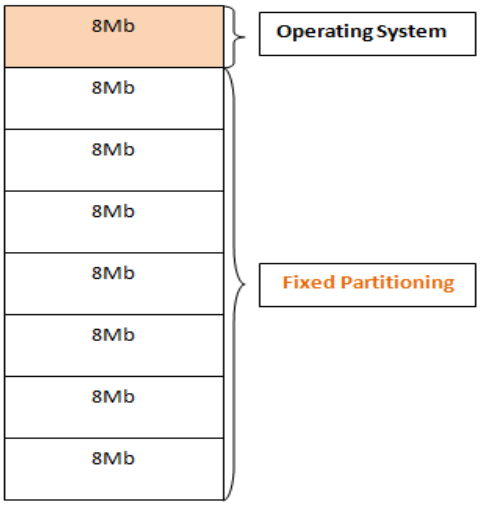
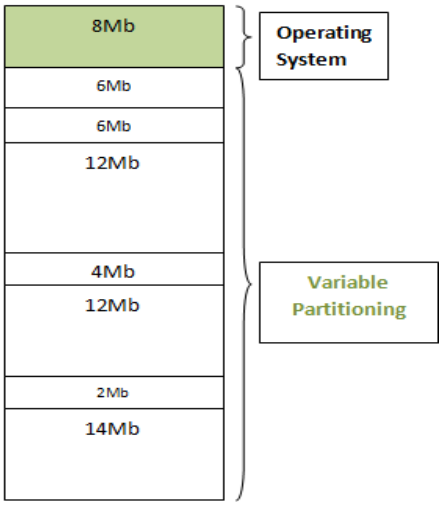


Figure 5: Variable Partitioning

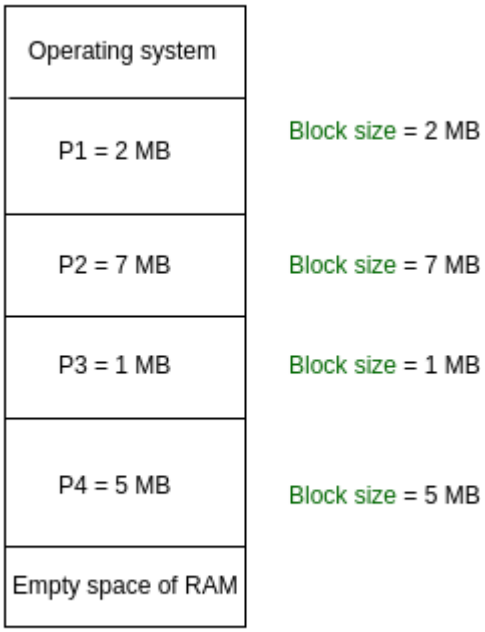


<u>Dynamic Multiple Partition Memory Management:-</u> (Variable partition)

- To overcome/eliminate some of the problems with fixed partition, another method is developedknown as "Dynamic Partitioning".
- In this technique, the amount of memory allocated is exactly the amount of memory a processrequires.

- In this method the partition are made dynamically.
- Initially when there is no process in the memory, the whole memory is available for allocationand it is treated as a single large partition of available memory (a hole).
- Whenever a process request for memory, the hole large enough to accommodate that process isallocated.
- The rest of the memory is available to other process.
- As soon as the process terminates, the memory occupied by it is de-allocated and can be usedby other process.

Dynamic partitioning



Partition size = process size
So, no internal Fragmentation

Advantage:-

1. Partition changed dynamically. So no internal fragmentation.
2. Efficient memory and CPU

utilization.Disadvantage:-

1. Suffers from external

fragmentation.Partition Selection Algorithms:-

Whenever a process arrives and there are various holes large enough to accommodate it, the operating system mayuse one of the following algorithm to select a partition for the process.

- First fit:- In this algorithm, the operating system scans the free storage list and allocates the first partition that is large enough for that process.

  Advantage:-

  1. This algorithm is fast because very little search is

  involved.Disadvantage:-

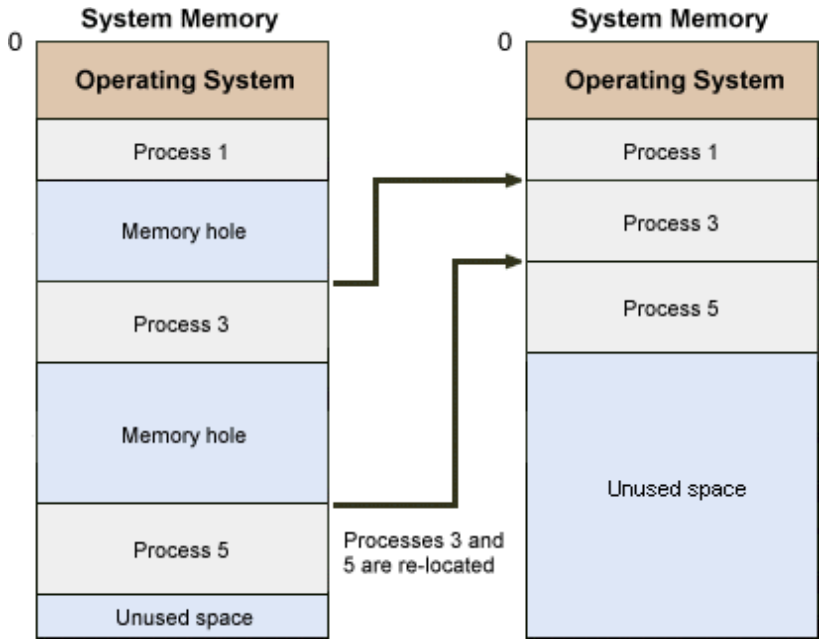
  1. Memory loss may be high.

- Best fit:- In this algorithm the operating system scans the free storage list and allocate the smallest partitionthat is big enough for the process.
  Advantage:-
  1. Memory loss will be smaller than the first fit.
  Disadvantage:- Search time will be larger as compared to first fit.
- Worst-fit:- In this algorithm the operating system scans the entire free storage list and allocate the largestpartition to the process.
  Disadvantage:-Maximum interval fragmentation.

Compaction:-

- Compaction is a technique of collecting all the free spaces together in one block, so that other process can use this block or partition.
- There are large no. of small chunks of free memory that may be scattered all over the physical memory and individual each of chunks may not big enough to accommodate even a small program.
- So, compaction is a technique by which the small chunk of free spaces are made contiguous to each other into a single free partition, that may be big enough to accommodate some other processes.

  Ex-Collect all the fragmentation together in one block and now the figure is:-



Non contiguous memory partition:-

- As one program terminates, the memory partition occupied by it becomes available to be usedby another program.
- Let the size of the freed memory be S, the next program to be run on the memory may need aspace which is larger or smaller than S.
- If it is larger then it cannot be loaded, if it is smaller, than a part of the partition remainsunutilized.
- This unutilized memory is known as fragment..This concept is known as fragmentation.
- Fragmentation is of 2 types:-
  - ✓ External fragmentation
  - ✓ Internal fragmentation.

External fragmentation

When the fragment is too small for a running program to be load, then there a fragment or portionremains unutilized.

Internal fragmentation

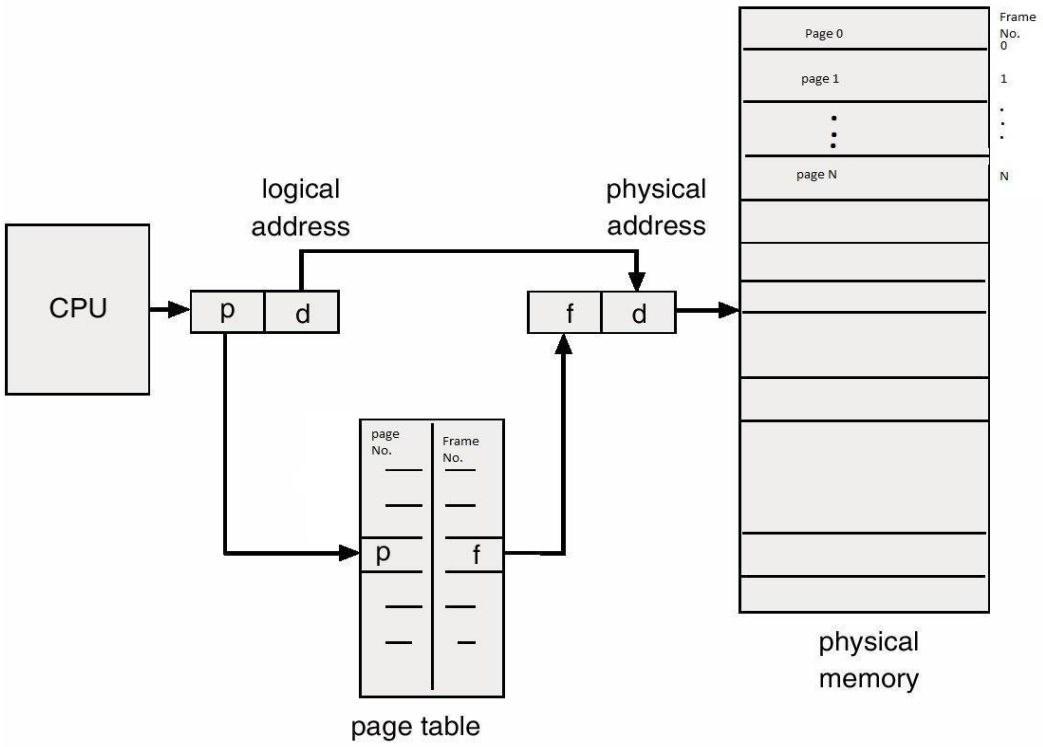
When the fragment remains unutilized inside a larger memory partition already allocated to a program.

- Both lead to poor memory utilization.
- To overcome this problem the memory is allocated in such a way that parts of a single process may be placed in non-contiguous areas of physical memory. This type of allocation is known as Non-contiguous allocation.
- The two popular schemes in Non-contiguous allocation are paging & segmentation.

Paging

- Paging is an efficient memory management scheme because it is Non-contiguous memory allocation method.
- The partition method supports the contiguous memory allocation i.e the entire process loaded in partition but in paging the process is divided into small parts, these are loaded into elsewhere in main memory.
- The basic idea of paging is physical memory/ main memory is divided into fixed size blocks calledas frames.
- Logical memory (or user job) is divided into fixed size block called pages.
- Page size and frame size should be equal.
- Backing store is also divided into fixed size block that are of same size as memory frames.
- When a process is to be executed its pages are loaded into the main memory in any availablememory frame.
- Every logical address generated by CPU is divided into two parts:-

1. Page number (P) 2.Page offset (d)



Structure of paging scheme

- Page number is used as an index into the page table.
- Page table is a data structure maintained by operating system. It is used for mapping purpose.
- The page table specifies-
  - Which frames are allocated
  - Which frames are available
  - How many total frames are there and so on.
- The page table consists of 2 fields- 1)**page number 2)frame number**
- page table contains the base address of each page in physical memory.
- The base address is combined with the page offset to define the physical memory address.
- The page size or frame size is depending upon operating system. But it is generally a power of 2,such as4MB,8MB,16MB etc
- The page map table specifies which page is loaded in which frame ,but displacement or offset iscommon.
- The paging has no external fragmentation, but there may be internal fragmentation .In paging it is called as **page break**.

Advantage

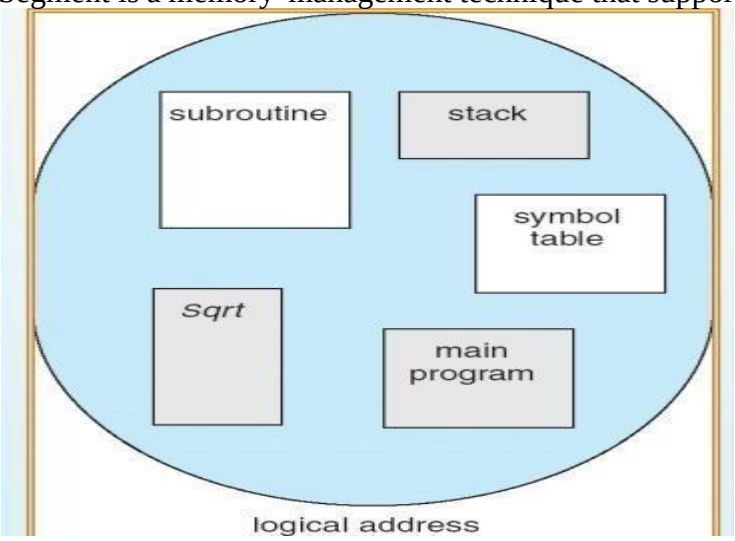
- It supports time sharing system
- It doesnot effect from fragmentation
- It support virtual memory.

Disadvantage

- The scheme may suffer “page break”.
- If the number of pages are high,it is difficult to maintain page table.

Segmentation

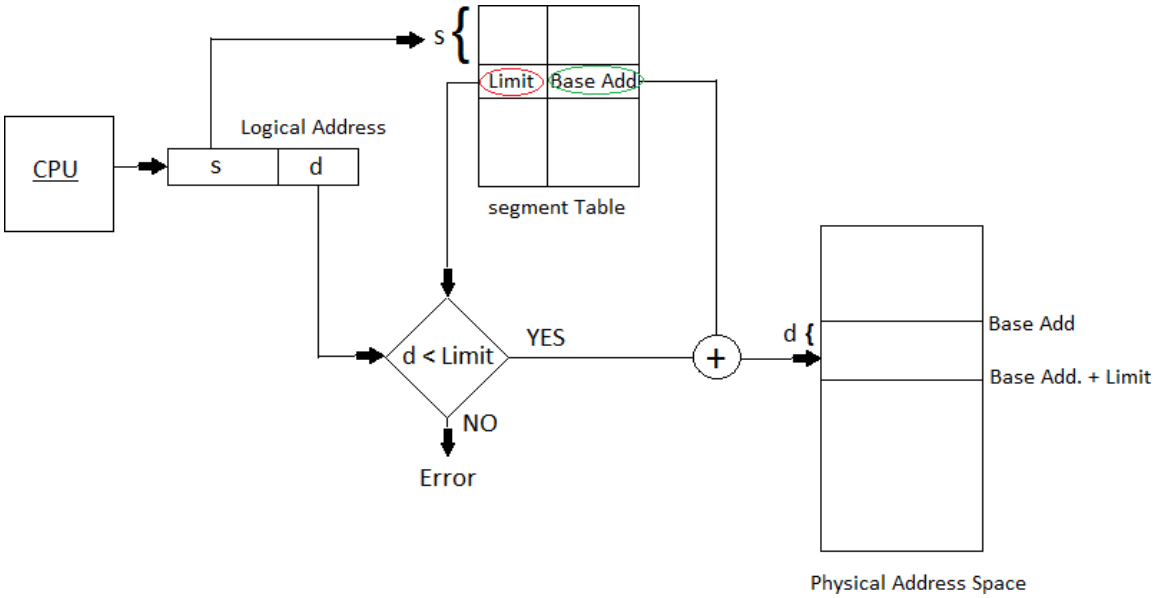
- In case of paging the user’s view of memory is different from physical memory.
- User don’t think that memory is a linear array of byte, some containing instruction and somecontaing data.
- But he view the memory as a collection of variable sized segments and there is no ordering ofsegments.
- Segment is a memory management technique that supports **user’s view of memory.**



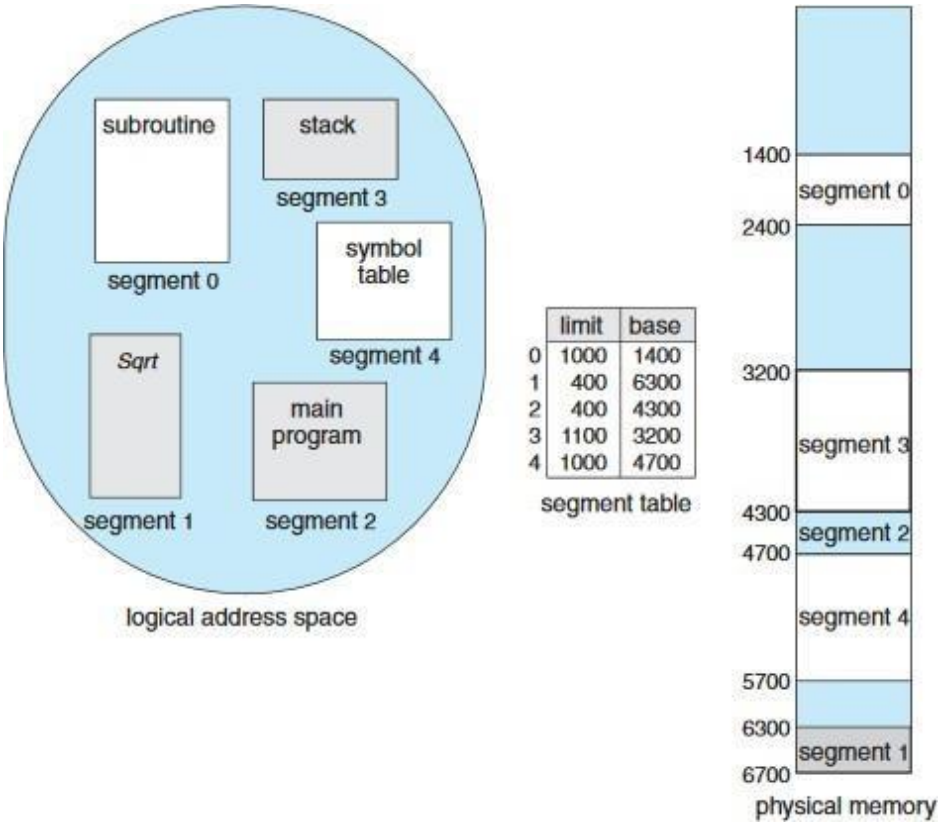
- A segment can be defined as a logical grouping of instruction ,such as subroutine, array or a dataarea.

- “Every program is a collection of these segments”
- Here the logical address is a collection of segment
- Each segment has a name and length.
- Segmentation is a technique for managing these segments.
- Each segment are numbered and referred by segment number.
- The logical address is consisting of two tuples<segment no, offset>
- Ex-the length of a segment main is 100K, here ‘main’ is the name of the segment and the offsetvalue is 100K.the operating system searches the entire main memory for free space to load a segment. This mapping is done by segment table.
- The segment table is a table, each entry of it has a segment “Base” and a segment “limit”.
- Logical address consist of two parts.
    1. Segment Number(s)
    2. Offset into that segment(d)
- The segment number is used as an index into the segment table.
- The offset is compared with segment limit.offset should be less than or equal to limit, else there is an error. If the offset is valid then “d” will be added with base value to get the actual physical address.

Diagram of segmentation scheme



Example:-



PAGE FAULT

- When the processor need to execute a particular page, that page is not available in mainmemory then an interrupt occurs to the operating system called as **page fault.**
- When the page fault is happened, the page replacement will be needed. The word page replacement means to select a victim page in the main memory.
- Replace the page with the required page from backing store or secondary memory.

STEPS FOR HANDLING PAGE FAULT

- To access a page the operating system first check the page table to know whether the reference is valid or not.
- If invalid, an interrupt occur to operating system called page fault.
- Then operating system search for free frame in the memory.
- Then the desire page is loaded from disk to allocate free frame.
- When the disk read is complete the page table entry is modified by setting the valid bit.
- The the execution of the process starts where it was left.

Difference. Between Paging and segmentation

Paging	Segmentation
The main memory partitioned intoframes or blocks	The main memory partitioned into segments.
The logical address space divided intopages by compiler or memory management unit.	The logical address space is divide into segments specified by the programmer.
It may suffer from page break orinternal fragmentation	This scheme suffer from external fragmentation
The operating system maintain page map table for mapping between frames and pages.	Segment map table is used for mapping.
It doesn't support user view of memory	It support user view of memory.
The processor uses page no. and offsetto calculate absolute address	The processor uses the segment no. and displacement to calculate the absolute address.

VIRTUAL MEMORY

Virtual memory is a technique which allows the execution of a process, even the logical address space is a greater than the physical memory.

Ex:let the program size or logical address space is 15 MB., but the available memory is 12MB. So, the 12MB is loaded in main memory and remaining 3MB is loaded in the secondary memory. When the 3MB is needed for execution then swap out the 3MB from main memory tosecondary memory and swap in 3MB from secondary memory to main memory.

Advantages:

Large programs can be written, as virtual space available is huge compared to physical memory.

Less I/O required, leads to faster and easy swapping of processes.

More physical memory available, as programs are stored on virtual memory, so they occupy very less space on actual physical memory.

DEMAND PAGING

Demand paging is the application of virtual memory.

It is the combination of paging and swapping.

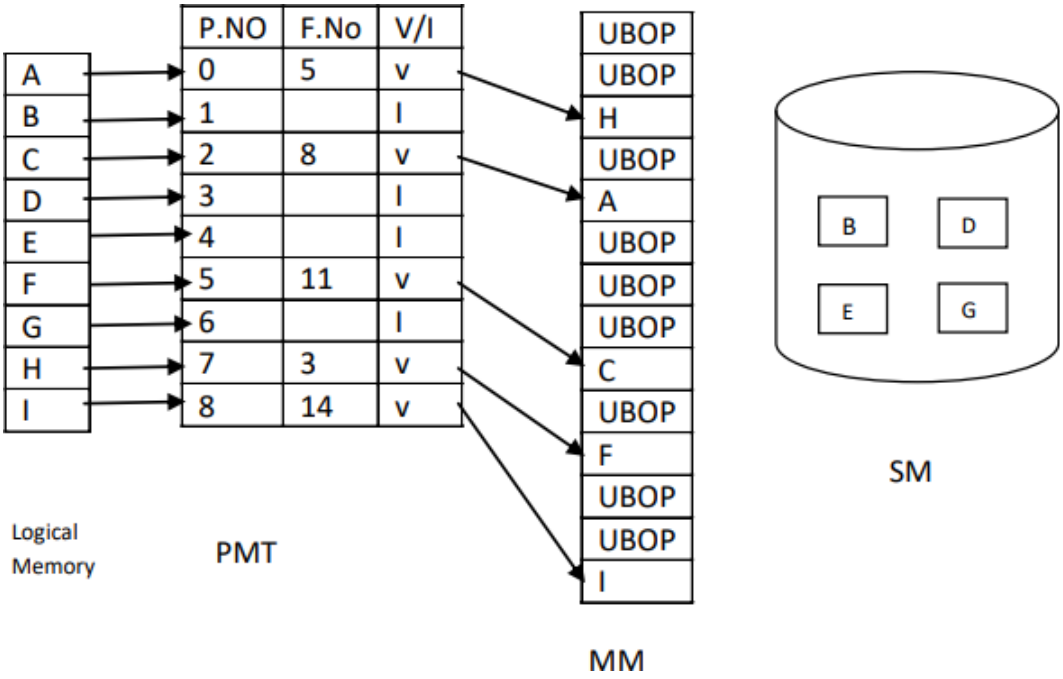
The criteria of this scheme is “a page is not loaded into the main memory from secondary memory,until it is needed”.

So,a page is loaded into the main memory by demand, so this scheme is called as “Demand Paging”.

For ex: Assume that the logical address space is 72 KB. The page and frame size is 8KB. So thelogical address space is divided into 9 pages, i.e numbered from 0to 8.

The available main memory is 40 KB.i.e.5 frames are available. The remaining 4 pages areloaded in the secondary storage.

Whenever those pages are required ,the operating System swap-in those pages into main memory.



In the above figure the mapping between pages and frames done by page map table.

In demand paging the PMT consisting of 3 fields i.e. page no., frame no. and valid/invalid bit. If a page resides in the main memory the v/I bit set to valid.

Otherwise the page resides in the secondary storage and the bit set to Invalid.

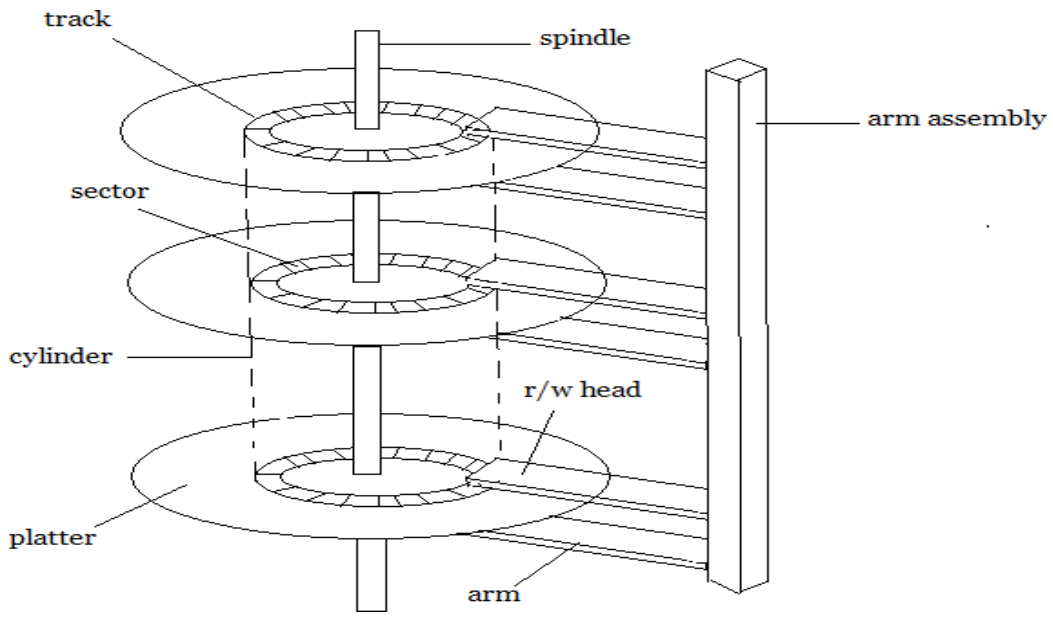
The page numbers 1,3,4,6 are loaded in the secondary memory. So those bits are set to invalid. Remaining pages resides in the main memory, so those bits are set to valid.

The available free frames in main memory is 5,so 5 pages are loaded, remaining frames are used by other process(UBOP).

UNIT-4
DEVICE MANAGEMENT

Magnetic Disk Structure:

In modern computers, most of the secondary storage is in the form of magnetic disks. Hence, knowing the structure of a magnetic disk is necessary to understand how the data in the disk is accessed by the computer.



Physical structure of a magnetic disc:-

Platter:- Each disc has a flat circular shape named as platter. The diameter of platter is from range 1.8^{11} to 2.25^{11} inches.

The information Are stored magnetically on platter.

Tracks:- Surface of the platter are logically divided into circular

tracks.**Sector:-** Tracks are subdivided into no. of sections termed as

sector. Each track may contain hundreds of sectors.

Cylinder:- Set of tracks that are at one composition makes up a cylinder.There

may be thousands of concentric cylinders in disc drive.

Read/write head:- This is present just above each surface of every platter.

Disc arm:- The heads are attached to a disc arm that moves all the heads as a unit.

Note:- The storage capacity of a normal disk drive is measured by GB.

Seek time:- The time required to reach the desired track by the read and write head is a seek time. There are 2 components to calculate seek time.

i) Internal start up time
ii) The time taken to traverse the cylinder
Ts = m + n + s
Where Ts = Estimated seek time
n = no. of tracks traverses
m = constant
s = start up time.

Rotational delays time:-

The time required to reach the desired sector by write/ read head is called rotational delay time. Generally the average of rotational delay is between 100 to 200 ms.

Disk scheduling algorithm:-

Whenever a process request an i/p, o/p operations from the basic it will issue a system call to the OS. The request have been processed according to some sequence. This sequence has been made by various algorithm named as disk scheduling algorithm
Disk scheduling algorithm Schedules all the request properly and in some order.
This algorithm. Is implemented in a multi programming system.
Following are the disk scheduling algorithm
i) FCFS
ii) SSTF (Shortest seek time first)
iii) SCAN
iv) C-SCAN (Circular scan)
v) Look
vi) C-Look

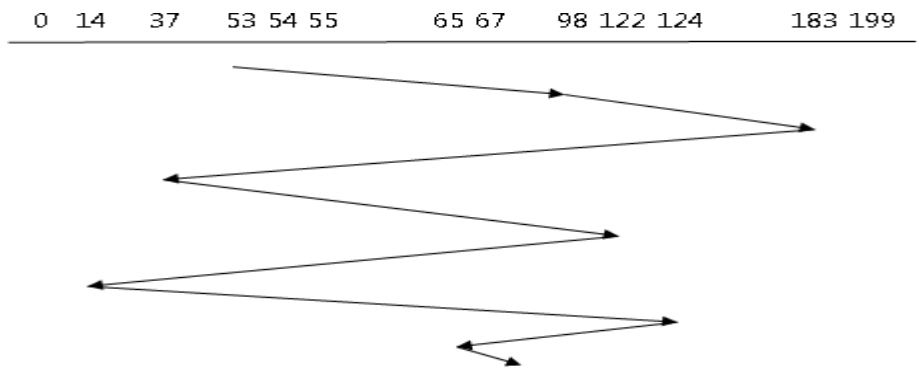
FIRST COME, FIRST SERVE

It is not an efficient algorithm but it is fair in scheduling the disk access.

For example, we are given a list of request for disk I/O to blocks on a cylinder.

98, 183, 37, 122, 14, 124, 65, 67

If the starting point is 53 then the access would be like below



The big jump from 183 to 37 could be avoided if somehow 14, 37 and 122, 124 are served together. This indicates the problem with the FCFS algorithm which is larger head movement.

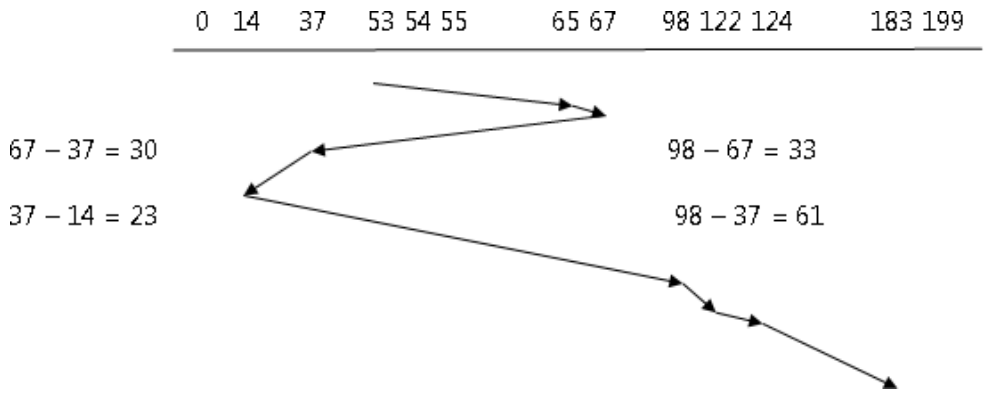
SSTF SCHEDULING

The main idea of the Shortest-Seek-Time-First algorithm is to service all the requests close to the current position of the head before moving far away to service other requests.

Example:

Considering our previous sequence of disk blocks access.

Queue = 98, 183, 37, 122, 14, 124, 65, 67



There is a substantial improvement compared to FCFS algorithm. The total head movement is as follows.

65 – 53 = 12 37 – 14 = 23 124 – 122 = 2

67 – 65 = 2 98 – 14 = 84 183 – 124 = 59

67 – 37 = 30 122 – 98 = 24

Total Head Movement = 236 cylinders.

But suppose 14 and 183 are a queue and a request near 14 came, it will be served and next one is also close to 14 came, it will be served first and this will lead to starvation of 183 in the queue.

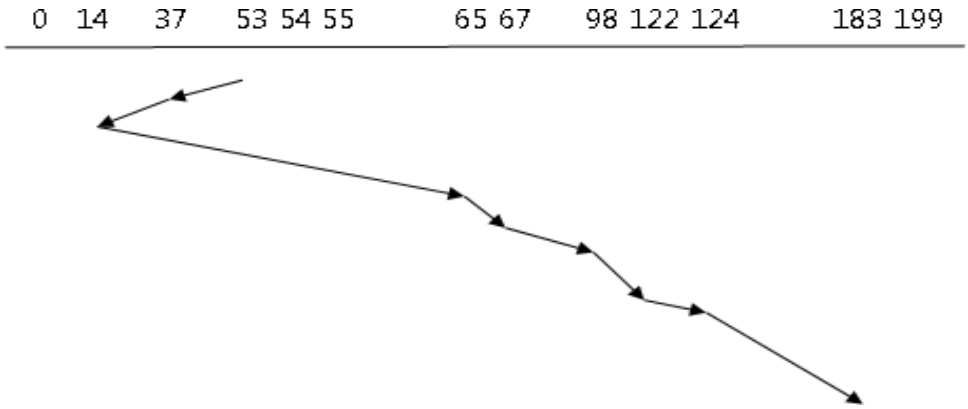
SSTF is an improvement but not the optimal algorithm.

SCAN SCHEDULING

In this algorithm, the disk arm works like an elevator starting at one end servicing all the way up to the other end and then start from the other end in reverse order.To use the SCAN algorithm,we need to know two information.

1. Direction of Scan
2. Starting point

Let's consider our example and suppose the disk start at 53 and move in the direction of 0.

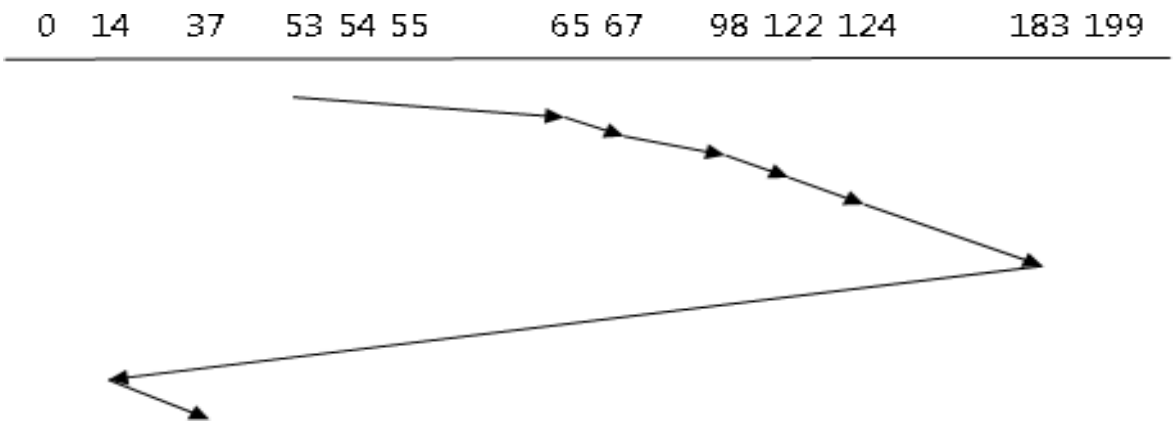


Total Head Movement

53 – 37 = 16 67 – 65 = 2 124 – 122 = 2
37 – 14 = 23 98 – 67 = 31 183 – 124 = 59
65 – 14 = 51 122 – 98 = 24
Total head movement = 158
The SCAN move in one direction and service all the request immediately, but while returning inreverse direction it does not serve any request since they have been serviced recently.
More of the request is at the opposite end, we will see an algorithm that wants to go the other end directly.

C-SCAN ALGORITHM

In this algorithm, the head from one end to the other servicing request along the way, however,it does not do a reverse trip and go to the beginning directly as if it is a circular queue.



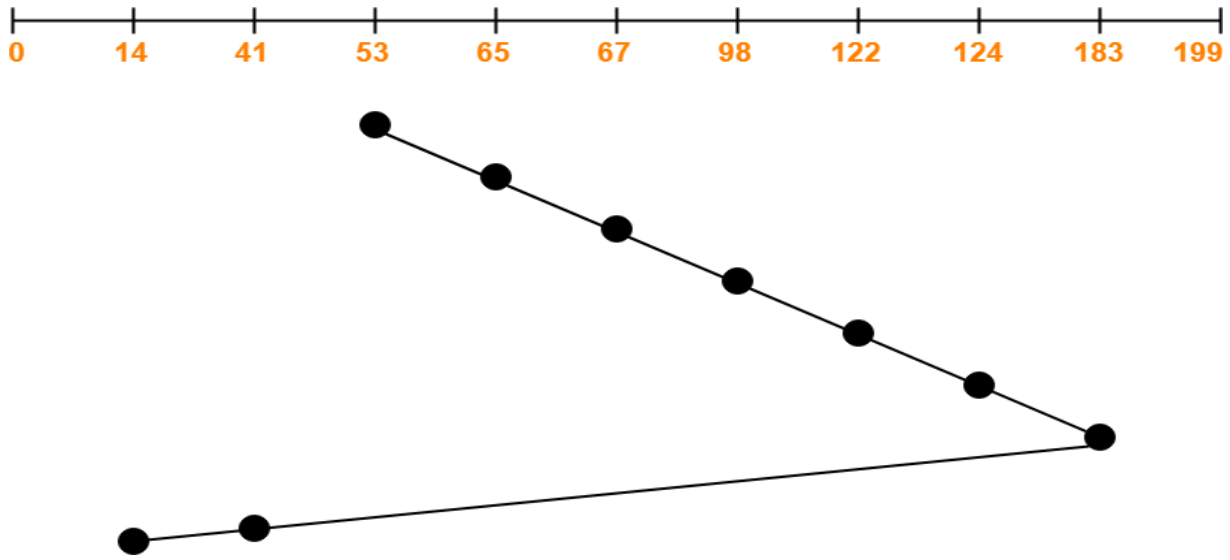
Total Head movement

183 -124 = 59 98 – 67 = 31 183 – 14 = Look for Request
124 – 122 = 2 67 – 65 = 2 37 – 14 = 23
122 – 98 = 24 65 – 53 = 12
Total Head Movement = 153

LOOK Disk Scheduling Algorithm-

- LOOK Algorithm is an improved version of the SCAN Algorithm**.**
- Head starts from the first request at one end of the disk and moves towards the lastrequest at the other end servicing all the requests in between.
- After reaching the last request at the other end, head reverses its direction.
- It then returns to the first request at the starting end servicing all the requests inbetween.
- The same process repeats.

Consider a disk queue with requests for I/O to blocks on cylinders 98, 183, 41, 122, 14, 124, 65, 67. The LOOK scheduling algorithm is used. The head is initially at cylinder number 53 moving towards larger cylinder numbers on its servicing pass. The cylinders are numbered from 0 to 199.



Total head movements incurred while servicing these requests

= (65 – 53) + (67 – 65) + (98 – 67) + (122 – 98) + (124 – 122) + (183 – 124) + (183 – 41) + (41 – 14)

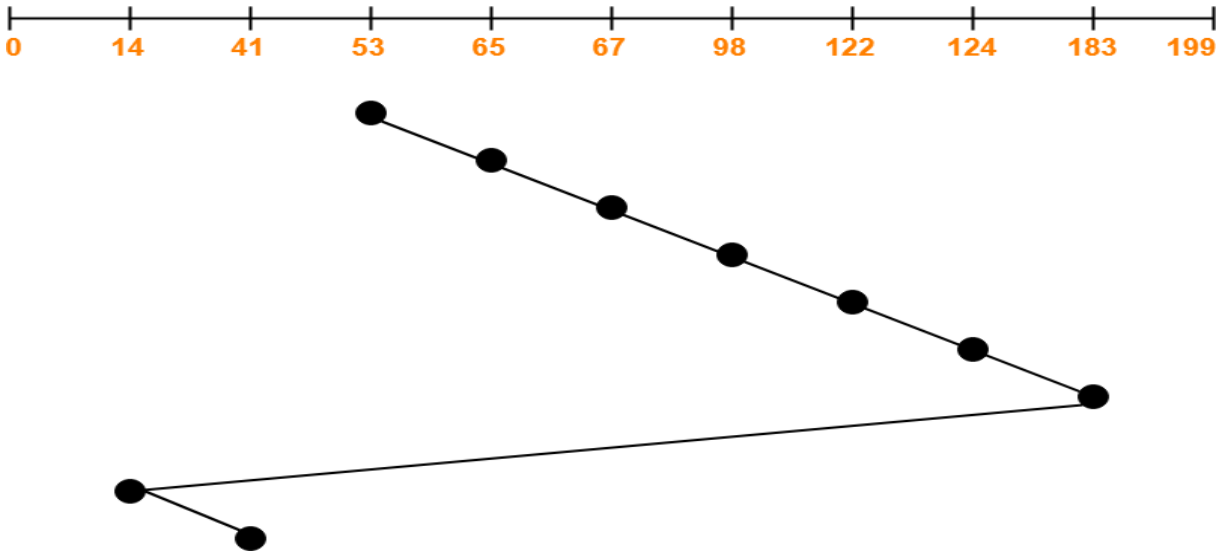
= 12 + 2 + 31 + 24 + 2 + 59 + 142 + 27

= 299

C-LOOK Disk Scheduling Algorithm-

- Circular-LOOK Algorithm is an improved version of the LOOK Algorithm.
- Head starts from the first request at one end of the disk and moves towards the last request at the other end servicing all the requests in between.
- After reaching the last request at the other end, head reverses its direction.
- It then returns to the first request at the starting end without servicing any request in between.
- The same process repeats.

Consider a disk queue with requests for I/O to blocks on cylinders 98, 183, 41, 122, 14, 124, 65, 67. The C-LOOK scheduling algorithm is used. The head is initially at cylinder number 53 moving towards larger cylinder numbers on its servicing pass. The cylinders are numbered from 0 to 199.



Total head movements incurred while servicing these requests

= (65 – 53) + (67 – 65) + (98 – 67) + (122 – 98) + (124 – 122) + (183 – 124) + (183 – 14) + (41 – 14)

= 12 + 2 + 31 + 24 + 2 + 59 + 169 + 27

= 326

Device management

The main functions of the device manager are:

1. Monitor the status of all devices, including storage drives, printers and other peripherals
2. Enforce pre-set policies on which process gets which device for how long
3. Deal with the allocation of devices to processes
4. Deal with the de-allocation of devices to processes, both at a temporary basis (e.g. when the process is interrupted) and on a permanent basis (e.g. when the process is completed).

Device management technique:-

There are 3 techniques for device management, i.e.

i) Dedicated.
ii) Shared
iii) Virtual

Dedicated:-

- These are devices that are assigned to one process at a time, and the process only releasesthe device once it is completed.
- The problem with this is that it means only one user is using it at a time, and it might be inefficient if the device isn't being used 100% of the time that it is being locked by the user.
- Ex.:-Printer, card reader and disk.

Shared:-

- These are devices that can be shared between several processes.
- Considering an example like a hard disk, it is shared, but interleaving between different processes' requests.
- All conflicts for device need to be resolved but predetermined policies to decide which request is handled first.

Virtual:-

- These are devices are a combination of Dedicated and Shared Devices.
- So a printer is a dedicated device, but using the spooling (queues) means it can be shared.
- A print job isn't sent straight to the printer, instead it goes to the disk (spool) until it is fully prepared with all the necessary printer sequences and formatting, then it goes to the printer,ensuring that the printers (and all I/O devices) are used efficiently.

I/O traffic controller:-

I/O traffic controller, control all the device track or channel.The traffic controller maintain all the status information.

The traffic controller, attend 3 questions, i.e.

i) Is there a path, available to server or I/O request?
ii) If more, than, one path available.
iii) If no path currently available, when, one will be free.

In order to answer these question, I/O traffic controller use one of the following data base, i.e.,

i) Unit control block (UCB)
ii) Central Unit Control Block (CUCB)
iii) Channel control Block (CCB)

UCB	CUCB	CCB
⇒ Device unit identification. ⇒ Status of device. ⇒ List of control unit, connected to the device. ⇒ List of processes, waiting for this device.	⇒ It is control unit identification ⇒ List of device connected to the control unit. ⇒ List of channel connected to the control unit ⇒ Waiting for the control.	⇒ Channel identification. ⇒ Status of the channel. ⇒ List of control unit connected to the channel. ⇒ List of the process, waiting for the channel

I/O scheduler:-

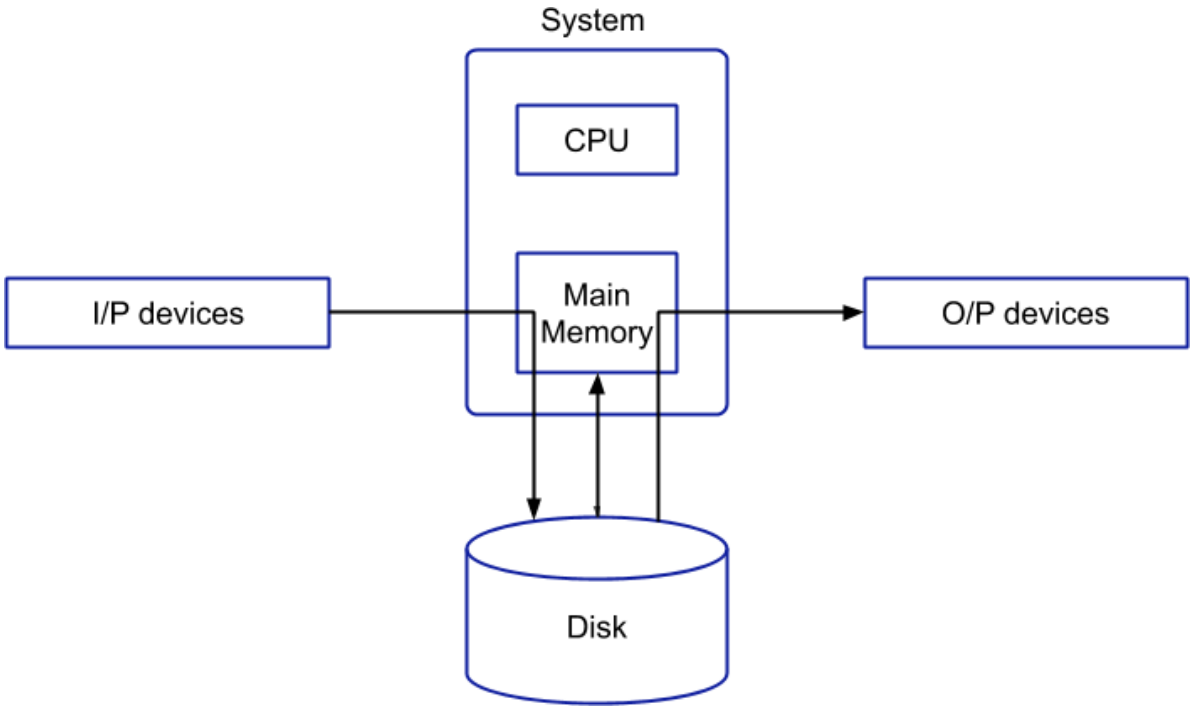
If there are more I/O request, pending, then, available path is necessary to choose, which, I/O request is satisfied, first. Then, the process of scheduling, is applied here, and it is known as I/Oscheduler.

I/O device handler:-

- I/O processes the I/O interrupts, handles error condition, and provides detailed scheduling algorithms, which are extremely device dependent.
- Each type of I/O device has own device handler algorithm like FCFS, SSTF, SCAN.

Spooling:

- SPOOL is an acronym for **simultaneous peripheral operations on-line**.
- It is a kind of buffering mechanism or a process in which data is temporarily held to beused and executed by a device, program or the system.
- Data is sent to and stored in memory or other volatile storage until the program or computer requests it for execution.



- Spooling uses buffer to manage files to be printed.
- Files which are spooled are queued and copied to printer one at a time.
- To manage I/O requests, operating system has a component that is called spooler.
- Spooler manages I/O requests to a printer. Spooler operates in the background and creates a printing schedule.

Race condition:-

Race condition is a situation, where, several process, access and manipulate, same data, concurrentlyand the execution depend on a particular order.

UNIT- 5

DEAD LOCKS

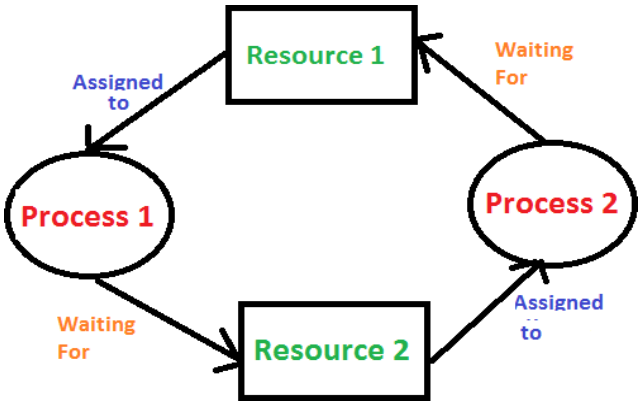
System Model:
A system consists of a finite number of resources to be distributed among a number of competing processes.The resources are partitioned into several types each of which consists of a number of identical instances. A process may utilized a resources in the following sequence

1) **Request:-**process request for a resource through a system call. If the resource is not available it will wait.
   Example: system calls open( ), malloc( ), new( ), and request( ).
2) **Use:-** After getting the resource, the process can make use of it by performing the execution.
   Example: prints to the printer or reads from the file.
3) **Release:-** After the completion of the task the resource is not required by that process, in that it should be released.
   Example: close( ), free( ), delete( ), and release( ).

Resources such as->CPU, memory, I/O devices etc. When a process requests for a resources then if it is free then it will be allocated to that process. But if the resource is busy with other process then the previous processhas to wait till that resource is free.

Deadlock: *Deadlock* is a situation where a set of processes are blocked because each process is holding aresource and waiting for another resource acquired by some other process.

For example, in the below diagram, Process 1 is holding Resource 1 and waiting for resource 2 which isacquired by process 2, and process 2 is waiting for resource 1.



REASONS/NECESSARY CONDITIONS FOR ARISING DEADLOCK:--

A deadlock situation can arise if the following four condition hold simultaneously in the system.

1. Mutual exclusion
2. Hold and wait
3. No pre-emption
4. Circular wait

1) **MUTUAL EXCLUSION:-** At least one resources must be held in a non-sharable mode. That meansonly one process can use that resource at a time.
EX: printer is non-sharable But HDD is a sharable resource.
So that if the resource is not free then the requesting process has to wait till the resource is released bythe other process.

2) **HOLD AND WAIT:-**There must be aprocess which is already holding using one resource andrequesting(waiting) for another resource which is currently held by another waiting process.

3) **NO PREEMPTION:-**Resources cannot be pre-empted. That means a resource can't released by the process unless until it has completed its task. I.e. printer will be released only when printing work is finished.

4) **CIRCULAR WAIT:-**Suppose there are n-processes {P0, P1, P2, ……Pn-1} they all are waitingprocesses.

P0 is waiting for the resource hold by P1.

P1 is waiting for the resource hold by P2.

P2 is waiting for the resource hold by P3.

– – – – – – – – – – – – – – – – – – – –

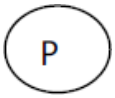
– –

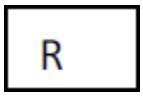
Pn-1 is waiting for the resource hold by P0.

If all above 4 conditions are satisfied in a system, then deadlock may occur but if anyone of the condition (criteria) is not satisfied then deadlock will never occur.

Resource allocation graph (RAG):-

- A diagrammatic represent to det. The existence of deadlock in the system by using a graph named as RAG.
- It is a directed graph.

- RAG consists of several no. of nodes and edges.
- It contains i) Process 

node Circle ii) Resource node 

Square.
- The bullet symbol within the resource is known as instances of that resource.
- Instance of resources means identical resources of same type.
- There exist 2 kinds of edges.

  i) Request edge.

  ii) Allocation / assignment edge

REQUEST EDGE:- Whenever a process request for resources then it is called a request edge.

- A request edge is drawn from the process to resource.

ASSIGNMENT EDGE:- Whenever a resource is allocated to a process the request edge is converted to an assignment edge from the instance of the resource to the process.

NOTE:-If the RAG contains NO CYCLE , then there is NO DEADLOCK in the system.

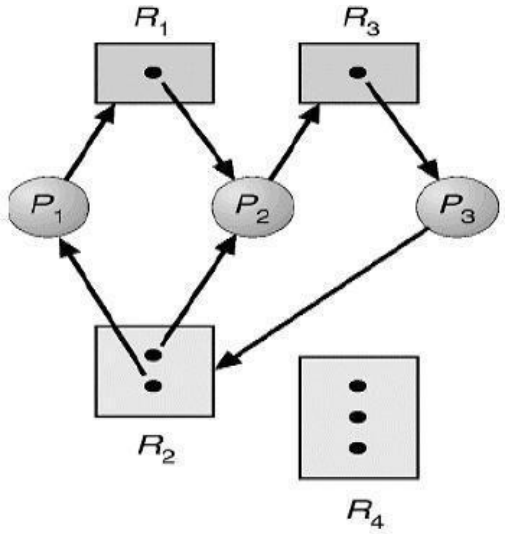
- If the RAG contains a CYCLE, then there MAY BE A DEADLOCK in the system.
- If the resources have exactly one instance then a cycle indicates a deadlock.
- If the resources have multiple instance per resource then a cycle indicates that "there may be deadlock".
- Process wait for graph(PWFG) :- A process wait for graph can be obtained by removing/collapsing resource symbols in the RAG.

The resource allocation graph shown in figure has the following situation.

- The sets P, R, E
- P = {P1, P2, P3}
- R = {R1, R2, R3, R4}
- E = {P1 → R1,P2 → R3,R1 → P2,R2 → P2,R2 →P1,R3 → P3}

The resource instances are

- Resource R1 has one instance
- Resource R2 has two instances.
- Resource R3 has one instance
- Resource R4 has three instances.



The process states are:

- Process P1 is holding an instance of R2 and waiting for an instance of R1.
- Process P2 is holding an instance of R1 and R2 and waiting for an instance R3.
- Process P3 is holding an instance of R3.

The following example shows the resource allocation graph with a deadlock.

- P1 -> R1 -> P2 -> R3 -> P3 -> R2 -> P1
- P2 -> R3 -> P3 -> R2 -> P1

Methods for Handling Deadlocks

Deadlocks can be handled by an of the following methods.

- Deadlock prevention
- Deadlock avoidance
- Deadlock detection and recovery

DEADLOCK PREVENTION: -Deadlocks can be prevented from occurring by preventing one of the necessary four conditions. I.e. mutual exclusion, hold and wait, no pre-emption and circular wait.
If one of the condition can be prevented from occurring then deadlock will not occur.

Eliminate Mutual Exclusion
It is not possible to dis-satisfy the mutual exclusion because some resources, such as the tape drive and printer, are inherently non-shareable.

Eliminate Hold and wait

1. Allocate all required resources to the process before the start of its execution, this way hold and wait condition is eliminated but it will lead to low device utilization. for example, if a process requires printer at a later time and we have allocated printer before the start of its execution printer will remain blocked till it has completed its execution.
2. The process will make a new request for resources after releasing the current set of resources. This solution may lead to starvation.

Eliminate No Preemption
Preempt resources from the process when resources required by other high priority processes.

Eliminate Circular Wait
Each resource will be assigned with a numerical number. A process can request the resources increasing/decreasing. order of numbering.
For Example, if P1 process is allocated R5 resources, now next time if P1 ask for R4, R3 lesser than R5such request will not be granted, only request for resources more than R5 will be granted.

DEADLOCK AVOIDANCE

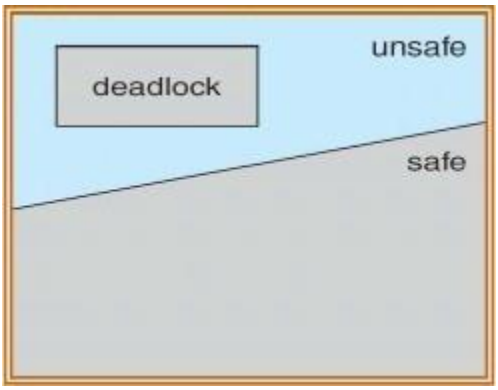
-> To avoid deadlock it is required to keep the additional information of the process i.e. the operating System will begiven prior information about the process such that

- Which process will request for which resource.
- At what time and for how long time.

-> So that the operating System find out a sequence of execution.

-> If all the process is execute in the sequence then system will never enter into a deadlock state.

SAFE STATE: - A state is safe if the system can allocate the available resources to each process in same order and still avoid deadlock.



*SAFE SEQUENCE: - A sequence of processes (P1, P2, and P3----------Pn) is a safe sequence if the available resources can be allowed to the processes in that sequence and avoid deadlock.

If no safe sequence exists the system is said to be unsafe state. An unsafe state may lead to a deadlock.

Ex. Suppose a system contains 12 tape drives

Process	max.	Allocation	need
P0	10	05	5
P1	04	02	2
P2	09	02	7

Free= 12-9 = 3

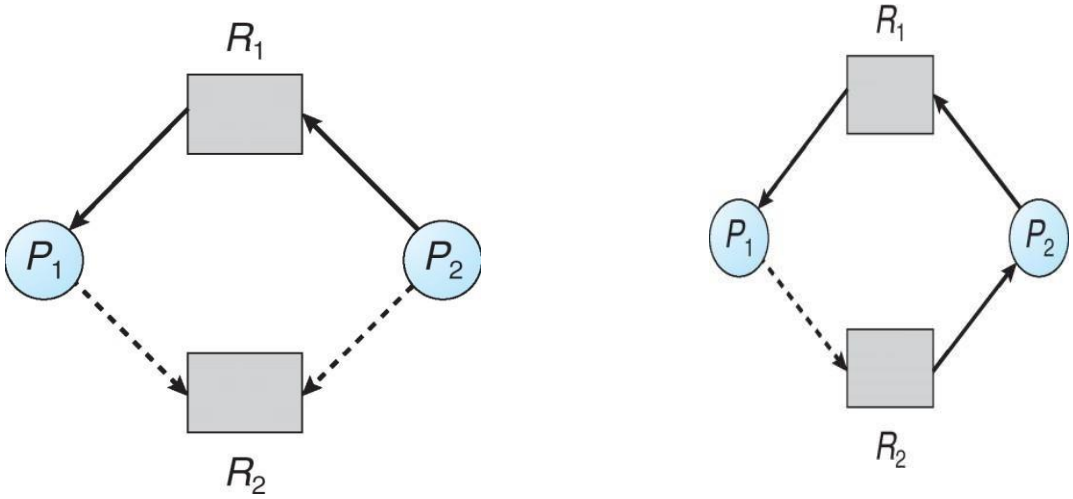
So, the safe sequence is <P1, P0, and P2>. If the system will always remain in safe state then deadlock will never occur. So, when a process requests for a resource that is currently available the system must decide whether that resource will be allocated or the process will wait. The request will be granted only if the allocation leaves the system in a safe state.

This method is used only when the system contain one type of resource having multiple instances.

Resource allocation graph :-(Multiple Resources Having single instance)

We can use this algorithm for deadlock avoidance if the system contains different types of resources but each is having single instances.

- In this graph, beside assignment edge and request edge, third edge known as "claim edge" isadded.
- A claim edge from process Pi to Ri indicates that the process Pi may request for resource Rj in FUTURE. Claim edge is similar to request edge but it is represented as dashed line,
- If a process request for resource Rj then that request only be granted if by converting the requesting edge Pi->Rj to the assignment edge Rj->Pi does not form a cycle.



- If there are two claimedges for the same resource then it can avoid. If only one of the processesis allocated the resource R1 then a deadlock can arise.

Banker's algorithm ;-(Multiple Resources having multiple instances)

The resource allocation graph allocation is not applicable to a resource having multiple instances of each resource type.

- This algorithm used for system having multiple resources along with multiple instances.
- Banker's algorithm is less efficient the RAG.
- This name was choose because it could be used in a banking system to ensure that bank never allocates its available cash such that it can no longer satisfy the needs of all its customers.
- When a new process enters into the system

  - It must declare maximum number of instances of each resource type that it may need.
  - This number should be less than the total number of resources.

- When a process requests a set of resources, the system must determine:-
  - Whether the allocation of these resources will leave the system in a safe state.
  - If YES, the resources are allocated to chat process.
- Is NO, the process have to wait until some other process releases enough resources.
- Banker's algorithm consists of two parts.
  - Safety algorithm
  - Resource request algorithm.
- The safety algorithm is used to determine whether a system is in safe state or not.
- The resource request algorithm is used determine whether or not a request generated bya process for a resource, would lead the system to an unsafe state.
- The algorithm uses several data structures such as vector & matrixes. Ex. Let in a system there are 'n' processes and 'm' resources.

1) **AVAILABLE**: - It is an array/vector of length 'm' indicates the number of available resource of each type.
If available [J] =k means there are k instances of resource type Rj available.
2) **MAX**: - It is am matrix defines the maximum demand of each process.
If max [i,j]=k, then the process Pi may request at most k instances of resource type Rj.
3) **ALLOCATION:** - It is an n*m matrix definesthe number of resources of each type currently allocated to each process. Ex. If allocation [i,j]=k, then process Pi is currently allocated k instances of resource to Rj.
4) **NEED**: - It is an n*m matrix indicates the remaining resources need each process if need [i,j]=k, then the process Pi may need k more instances of resource type Rj to complete its task.
Need[i,j]=max[i,j] — allocation[i,j].

Safety algorithm

The algorithm for finding out whether or not a system is in safe state. It can be described as follow.

STEP 1:- work is a vector of length
m.Finish is a vector of length n.

Work=
available.
Finish[i] =false.
For i=1, 2, 3 ----------n.

STEP 2:- find an i such that
Finish[i] =false.
Need i <=
work.
If no such i then go to step 4.

STEP 3:- work = work

allocationFinish[i] = true.

Go to step 2.

STEP 4:- If finish[i] = true for all I, then system is in safe state.

- The complexity of the algorithm is 0(m*n^2)i.e. the algorithm may require on order of m*n^2 operation to decide whether a state is safe.

Resource request algorithm

Request is a vector of length m.

1. If request i<= need i go to step-2.
   Else raise an error condition.
2. If request < available go to step-3.
   Otherwise Pi must wait.
3. Available = Available - Request(i)
   Allocation = Allocation + RequestiNeed i= need i-Request i

DEADLOCK DETECTION

If a system does not employ either a deadlock prevention or a deadlock a avoidance algorithm, thena deadlock situation may occur.

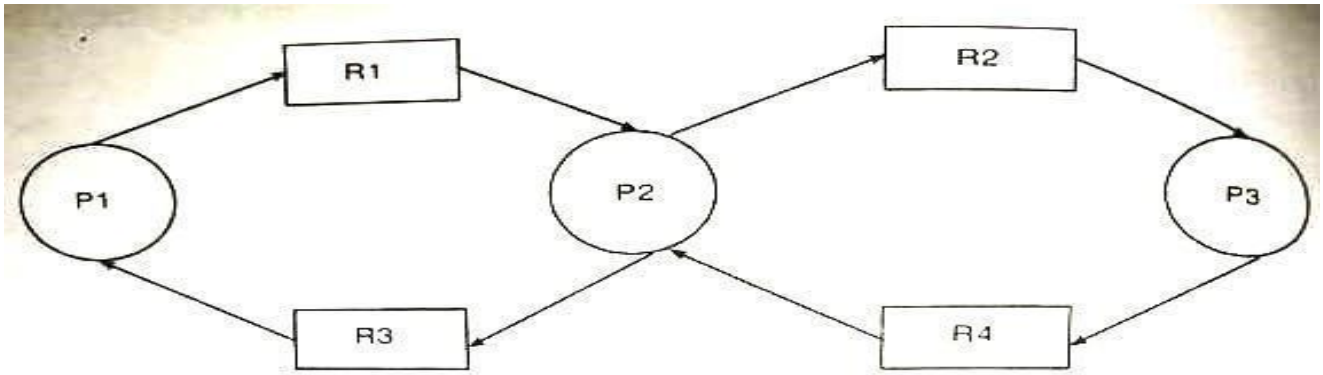
- So at the time or environment the system should provide:-
- An algorithm that will check whether the deadlock has occurred into the system (deadlock detection).
- An algorithm to recover the system from deadlock. State (deadlock Recovery)

Single instance of each resource type:-

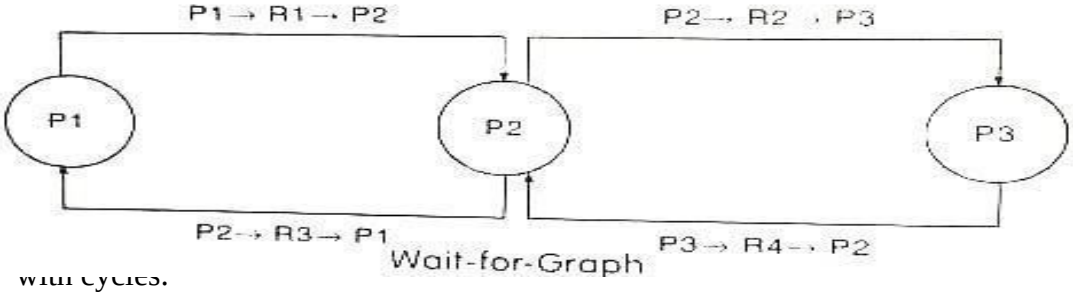
If all the resources have only single instance, then we can detect a deadlock state by using "wait-for-graph" (WFG).

It is similar to RAG but only difference is that here the vertices are only processes.

- There is an edge from Pi to Pj if there is an edge from Pi to R and also on edge from R to Pj.



Resource allocation Graph



Wait-for-Graph

dlocks will cycles.

- In this fig. there is two cycles one is P1 to P2 to P1. Second are P2 to P3 to P2.So the system consisting of two deadlocks.

<u>Multiple/several instances of Resource Type</u>:-

The wait for graph method is not applicable to several instance of a resource type.

- So, we need another method to resolve this problem. The algorithm used is known as "deadlock detection algo".
- This algorithm like "banker's algo" and it uses several data structures.
  -Available: - A vector of length 'm' indicates the number of available resources of each type.
  -Allocation: - An n*m matrix defines the no. of resources of each type currently allocated to each process.

-Request: - An n*m matrix indicates the current request of each process. If request [I,j]=k, then the process Pi is requesting k more instances of resource type Rj.

Detection algorithm:-

STEP 1:-

(work=available)

Available I, for i=1, 2, 3, 4---------

If Allocation i! = 0

Then finish[i]

=false

Otherwise finish[i] =true.

STEP 2:- Find an index such that

B finish[i] =false

Ij request[i] <=available or

workFinish [i] = true

Go to step-2

STEP 4:- If finish [i] =false, for some i, then the system is in deadlock state. I.e. process Pi is deadlocked.

RECOVERY FROM DEADLOCK

When the detection algorithm detects that deadlock exists in the system then there are two methods for breaking a deadlock.

- One solution is simply to abort one by one process to Break the circular wait.
- Second solution is to pre-empt some resources from one or more of the deadlock process.

⇨ **PROCESS TERMINATION:-**

This method used to recover from deadlock. We use one of two methods for process termination.

- Abort all deadlocked process
- Abort one by one process until all cycle is eliminating.

i. **Abort all deadlock processes: -** It means that release all the processes in the deadlocked state and start the allocation from the starting point.

- It is an expensive method.

ii. **Abort one by one process until the deadlock cycle is broken**:- In this method first abort the one of the processes in the deadlocked state and allocated the resources (resources from abort process) to some other process in the DL state.

- Then check whether the deadlock braked or not.
- If YES, then it is ok i.e. deadlock is eliminated. If NO, abort the process from the deadlock state then check.
- Continue this process until we recover from deadlock.
- This also expensive method but better than first one.

- In this method there is an overhead because every time the DL detection algorithm is invoked after each process is aborted.
- Ex. End task in windows.
- There are some features which determines the which process to be aborted::
    I. Priority of the process.
    II. How long the process is computed and how long time it is needed to completion.
    III. How many resources the process has currently used.
    IV. How many more resources it needs for completion.

⇨ **RESOURCE PREEMPTION:-** There are three methods to eliminate deadlocks using resourcepre-emption. They are-

- Selection a victim.
- Roll back
- Starvation

Selecting a victim:- Select a victim resource from the DL state, and pre-empt that one.

- Selection of victim done so that the cost will be minimum.

Rollback: - When a resource will be pre-empted from a process, then naturally the process will go into the waiting state. So we must roll back the process to some safe state so that it will be started fromsame state again or not from the beginning. I.e. roll back the processes and resources up to some safe state, and restart it from that state.

- This method requires the system to keep more information about the state of all the running processes.

Starvation: - How to ensure that starvation will not occur? It should be kept in mind that resources should not be pre-empted from etc. same process again and again; otherwise that process will not be completed for a long period of time.

- That is a process can a resources can be picked as a victim only a finite number of times, not morethan that, otherwise it create a starvation.

Unit-6

File Management

File is a primary resource in which we can store information and can retrieve theinformation when it is required.

There can be a numeric data file, an alphabetic data file or alphanumeric and binarydata files. In general terms a file is sequence of bits, bytes, lines or records.

All computer applications need to store and retrieve information. As computers canstore information on various storage media, in the same way, the operating System provides a logical view of information storage on various secondary storage media like magnetic disks, magnetic tapes and optical disks etc.

This uniform logical storage unit is called as file. So a file is the collection of related information, which is stored on secondary storage.

FILE ATTRIBUTES

A file has different attributes. The attributes may vary from one operating System to other.

*Name- A name is usually a string of characters. A symbolic name which is in human readable form.

*Identifier- it is usually a number and is a unique tag that identifies the file within the file system.it is a unique identification of a file which is internal to the system.

*Type- Normally expressed as an extension to the file name. It indicates the type of file.

Ex: .exe-executable file, .src-source file, .obj-object file

*Size-the current size of the file(in bytes)

*Location-it is a pointer to the location where a file is stored in secondary memory.

*Protection-It specifies the access control information .it controls who can do reading ,writing, executing and so on.

*Time and Date-it specifies time of creation and file created date.

*User identification-this is useful for protection and security and last usage monitoring.

File System

The file system consists of 2 distinct sub components.A

.collection of files, each storing related data

B. Directory structure, which organizes and provides information about all the file in thefile system.

FILE ORGANIZATION

File organization refers to the manner in which the records of a file are organized on the secondary storage.

Basically file is a set of logical records . It is allocated a disk space in terms of physical blocks.

The most common file organization schemes are:

- Sequential
- Direct
- Indexed
- Partitioned

Sequential:-In this method, information or record that is stored in a file is processed ina sequence .i.e the records are stored strictly in the same order as they occur physically in the file.

Direct:- The records are stored in any order as suited for application.The

system supports random or direct access of any record in the file.

Indexed:- In this method, an index is created for the file. This index contains thepointers(physical address) for various blocks or records.

Partitioned:- In this method,file is partitioned into sequential sub files.Each

sequential sub file is called a member of the partitioned file.

FILE OPERATION

To define a file in a proper manner ,there are different operations are performed on files.

To allow storage and retrieval of information from a file different system provide different operations.

The most common operations that can be performed on a file as follows:

***create-** 2 steps are needed to create a file.

-check whether the space is available or not.

-if yes,2nd one is made an entry for new file.

*__write-__ To write a file, we have to know 2 things

i) Name of the file.

ii) Information or data to be written on the file.

The system first searches the entire given locations for the file.if the file is found,the system must keep a pointer to the location in the file where the next write is to take place.

*__Read:-__ To read a file ,first of all we search the directories for the file.

If file is found, the system needs to keep a read pointer to the location in the file where the next read is to tae place.

Once the read has taken place,the read pointer is updated

*__Seek:-__ To reposition the file pointer to specified location. This is

done to read or write a record at a specified position.

*__Delete:-__ When the file is no longer required ,it is needed to delete the occupied diskspace.

To delete a file ,we search the directory for the named file and when the file is found ,we release all file space so that other files can reuse this space and erase the directory entry.

Truncate:- When the user wants to erase the contents of a file, but wants to retain it's attribute. It is not necessary to delete the file and then recreate it. It is possible by doing truncate operation. i.e to truncate a file , remove the contents only, but the attributes areas it is.

Open:- A process open must open a file before using it.

Close:- When all the accesses are finished, the attributes and disk addresses are nolonger needed, you need to close the file in order to release the internal table space.

Append:-

- This operation is restricted form of write.
- It only allows you to add data to the end of file.

Rename:-

- It frequently happens that a user needs to change the name of an existing file.
- This operation allows you to rename an existing file.

FILES TYPES

When designing a file system, we need to consider whether or not the operating system would recognize and support file types. A common technique for implementing file types is to include the types as the part of the file names.

- Generally , the name of the file split into two parts:- 1-name 2-extension (which is usually separated by 0).
- The file type is depending on extension of the file.
- The following section describes different types of files with their extension and function

File type	Extension	Purpose/function
executable	.exe .com .bin .none	Ready to run or ready to run m/c language
Object files	.obj	Instructions are in the form of m/c language. A linked uses this information and coverts itinto executable format.
baten	.bat .sh	Commands to the command interpreter.
Source code	.c .cc .cpp .java .pas .asm .f77	Source code in various language.
text	.txt .doc	Used to create text documents.
Word processor	.wp .rtf .etc	These file allows various word processor formats.
library	.lib .a .so .dll	Explain the entire library functions in any program.
Print or view files	.ps .pdf .jpg .dvi .sif	ASCII or binary files in a format printing or viewing.

archive	.arc .zip .tar	Grouped files , compressed file archivingor storage
multimedia	.mov .mpeg .mp3 .ym .mp4 .avi	These are binary files that contains audio/alu information.

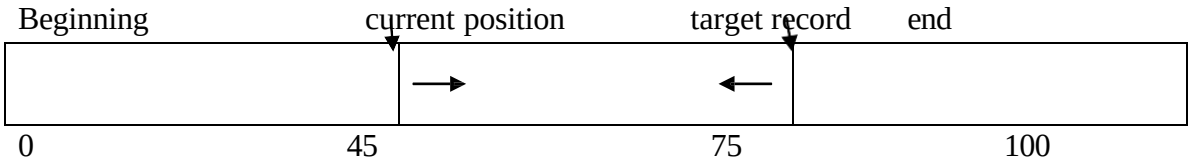
FILE ACCESSING METHODS

Files are used to store data. The information present in the file an be accessedby various access methods. Different system uses different access methods
.Following are the most commonly used access methods:

- Sequential access
- Direct access
- Indexed sequential access.

Sequential access method:-

- This method is simplest among all methods. Information in the file is processedin order, one record after the other.
- Magnetic tapes are supporting this type of file accessing
- Ex-a file consisting of 100 records , the current position of read/write head is 45threcord, suppose we wants to read the 75th record, then it access sequentially from 45,46 70,71,72,73,74,75.
- So, the read/write head transverse all the records between 45 to 75.



- Sequential files are typically used in batch application and parallel processing.

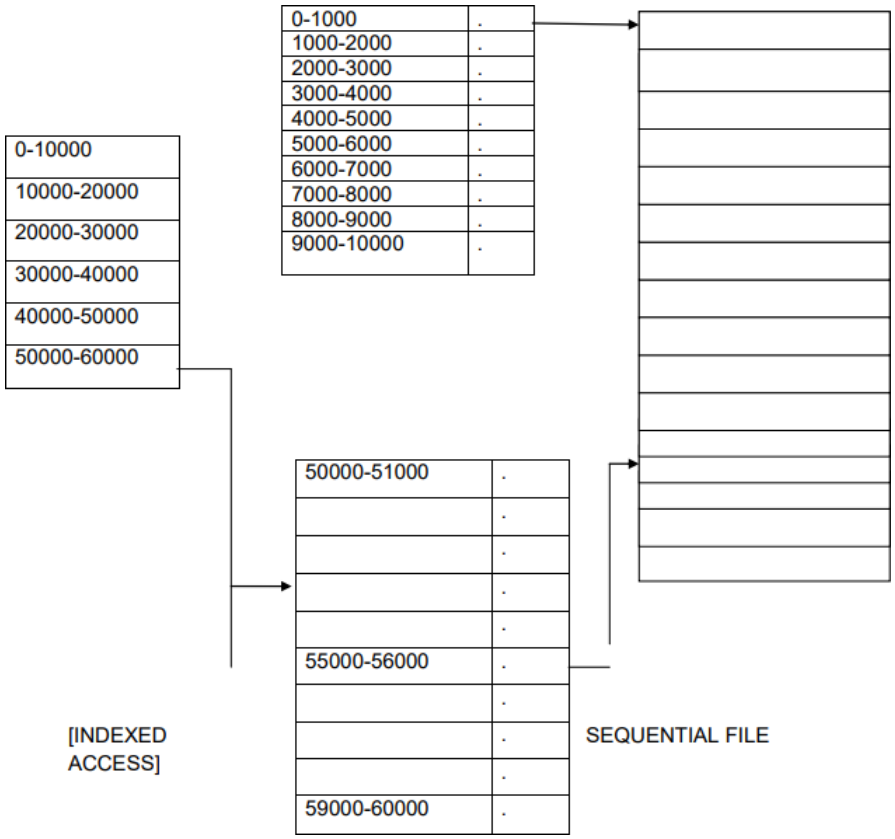
Direct access:-

- Direct access is also called relative access. In this method records can read/write randomly without any order.
- The direct access method is based on disk model of files because, diskallows random access to any file block.
- Ex:- a disk containing of 256 blocks. The current position of r/w head is 55th block, suppose we want 200th block. Then we can access 200th block directly without any restrictions. Another example is suppose a CD containing 10 songs, at the present we are listening the song no.3 and wewant to listen song no. 7, then we can shift from song no.3 to 7 without any restrictions.

Indexed sequential Access:-

The main disadvantage in the sequential file is , it takes more time to access arecord. To overcome this problem, we can use this method.

- In this method (ISF), the records are stored sequentially for efficient processing.But, they can be accessed directly using Index or key field. Keys are the pointerwhich contains adder of various blocks.
- Records are organized in sequence based on a key field.



- Suppose a file consisting of 60,000 records, the master index divided thetotal index into 6 blocks.
- Each block consisting of a pointer to secondary index.
- The secondary index divide the 10000 records into 10 indexes.
- Each index consisting of a pointer to its original location .1- A key field
  2- A pointer field

- Suppose we want to access the 55,550th record, then the file managementsystem (FMS) access the index that is 50000 to 60000.
- This block (50000 to 60000) consisting of a pointer, this pointer points to the 6thindex of the secondary index.
- This index points to the original location of the records from 55000 to 56000.
- From this it follows the sequential method
- That's why is method is said to be indexed sequential file. so this method isneither purely sequential nor purely direct access.
- Generally indexed files are used in air line reservation system and payrollsystem.

File directories:-

- The directory contains information about the files including attributeslocations and ownership.
- Sometimes the directories consisting of sub-directories also.
- The directory is itself a file and it is owned by the operating system.
- It is accessible by various file management units.

Directory structure:-

Sometimes the file system consisting a millions of files when no. Of files increases, then it is very difficult to manage the files.

- To manage these files:-
  - First group these files.
  - Then load one group of file in to one partition.

- This each partition is called "directory".
- A directory structure provides a mechanism for organizing many files in thefile system.

  Different operations on the file directories:-

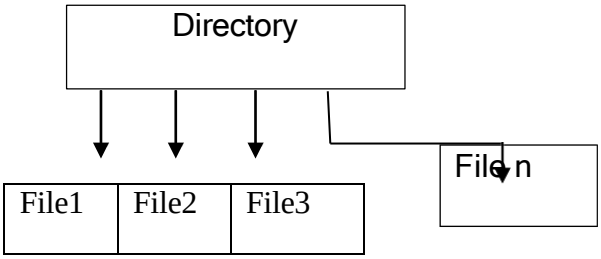
- Search for a file:-search the directory structure for required file.
- Create a file:- whenever we create a file then we should make an entry in the directory.
- Delete a file:- when file no longer needed, then we remove is from the directory.
- List a directory:-we can see the list of files in the directory.

- Rename a file:- whenever we want to change the name of file then we canchange.
- Traverse a file:- if we need to access every directory and every file with in the directory structure we can traverse the file system.

There are different types of directory structures are available they are:-

- Single level directory
- Two level directory
- Tree structured directory
- Acyclic directory
- General graph directory

- Single level directory:-
  - It is simplest of all directory structure.
  - In this directory system , only one directory is there and is consist of allfiles
  - Al files contained in the same directory name.



Advantage:-This scheme is very simple and ability to locate files easily.

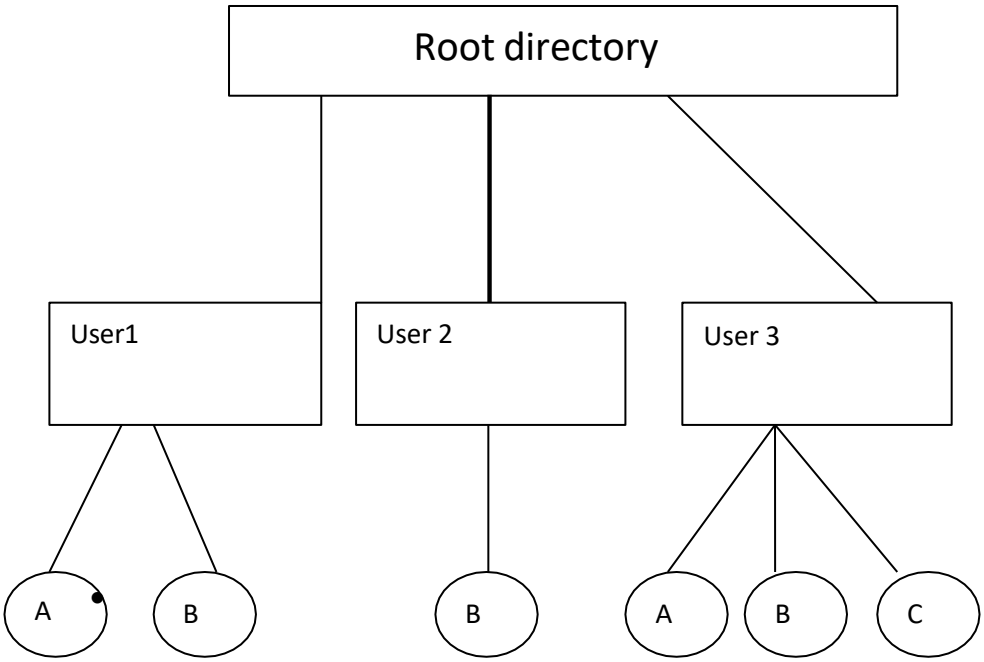
Disadvantage:-

- This structure have some significant limitations even for a single user, because if the no. Of files increases, then it is difficult to keep track of the file and also quite difficult to remember the names of all the files.
- As these files are in same directory, therefore these files will have unique names.

Two-level directory:-

- The problem is single level directory is different users may be accidentally usethe same name for their files.
- To avoid this problem each user need a private directory, so that name chosenby one user don't interface with the name chosen by different user.
- Two-level structure is divided into two levels of directories:-1-
  master directory (root directory)
  2-sub-directory (user directory)

- Consider the following figure for better understanding:-

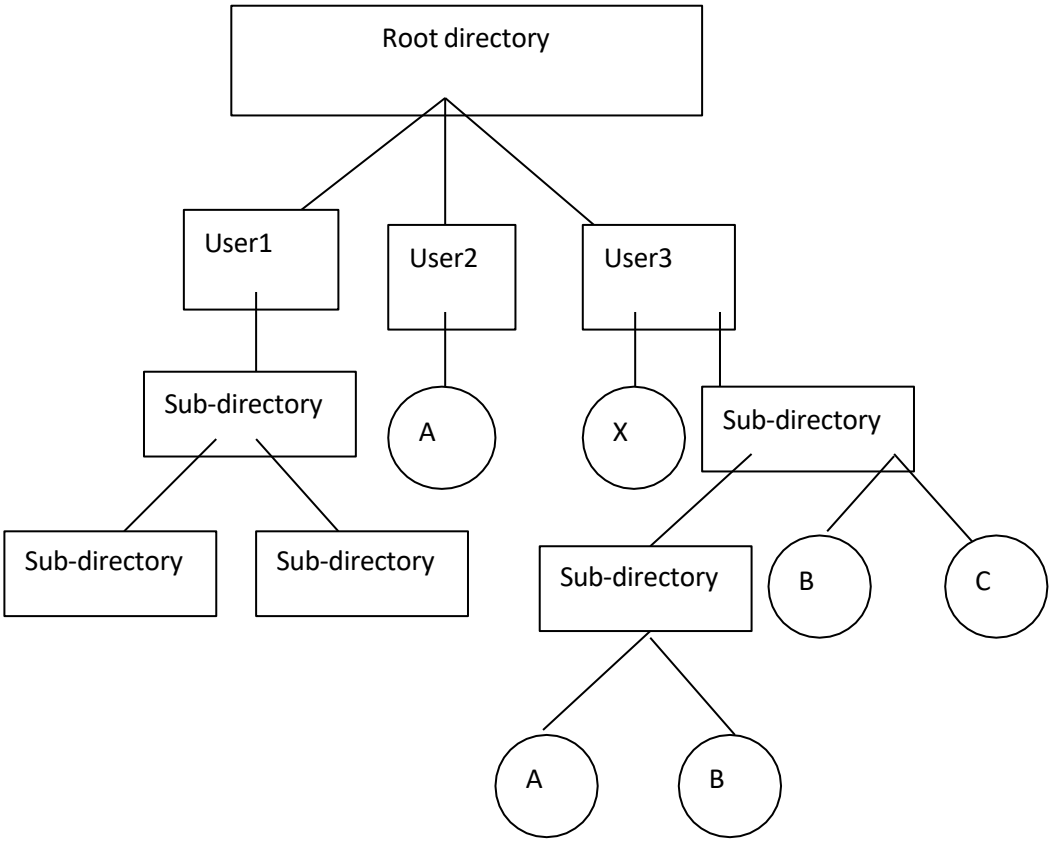


- Here root directory is first level directory. It consists of entries of ”user directory”.
- User level directories are user1, user2, user3 and it contains A, B, C files.

Tree structured directory/hierarchical directory system:-

- This structure allows user to create their own sub-directories and then organizethe files into it.
- MS-DOS operating system uses this tree structured directories.
- One directory may contain another directory as well as files .

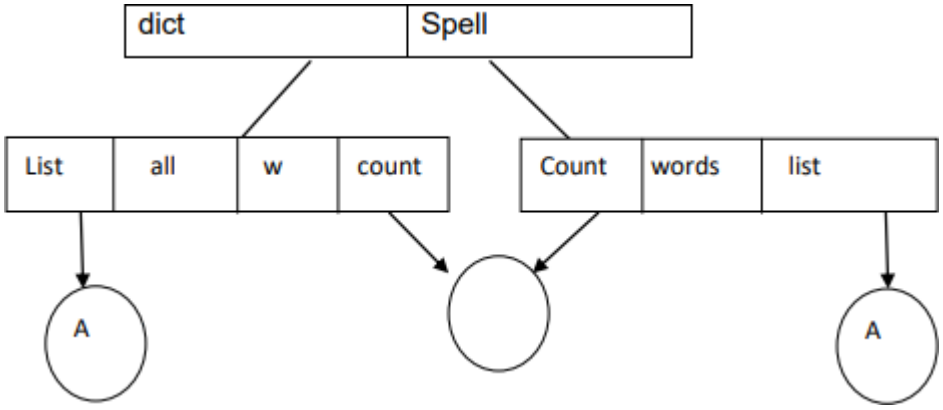
- Consider the following figure:-



[a tree structured directory]

Acyclic graph directory:-

- This graph allows directories to have shared sub-directories and files.
- Same file may be in two different directories.
- Acyclic graph is a generalization of the tree structured directoryscheme but here cycle informed.

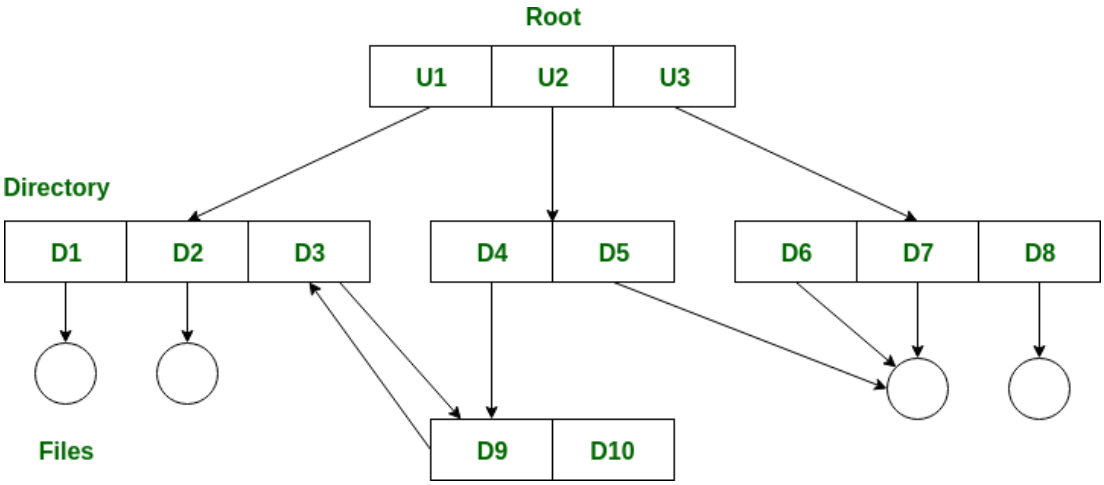


File implementation:-

- Shared files/sub-directories can be implemented in two ways
  - Symbolic link:-
    - A pointer to another file or directory. Ex-ln-s/ spell/ count/ dict /count.
  - Hard link:-
    - Duplicate all links information about them in both sharing directory.

General graph directory structure:-

- When we add links to an existing tree structured directory the tree structure is destroyed, resulting a simple graph structure.
- The primary advantage of this structure is traversing is easy and file sharing also possible.



File allocation method:-

Files are normally stored in the disks so the main problem is how to allocate to thesefiles so that disk space is utilized effectively and files can be accessed quickly.

- Three major methods of allocating disk space are in wide use
- They are:-
  1-contiguous allocation2-
  linked allocation
  3-grouped allocation or indexed allocation

Contiguous allocation:-

- In this method each files occupies a set of contiguous blocks on the disks.
- Ex:- a disks consisting of 1Kb blocks. A 100kb file would be allocated to 100 consecutive blocks. With 2kb blocks, it would be allocated 50 consecutive blocks.

Directory

file	start	length
count	0	2
tr	14	3
mail	19	6
list	28	4
f	6	2

The *file 'mail'* in the above figure starts from the block 19 with length = 6 blocks. Therefore, it occupies *19, 20, 21, 22, 23, 24* blocks.

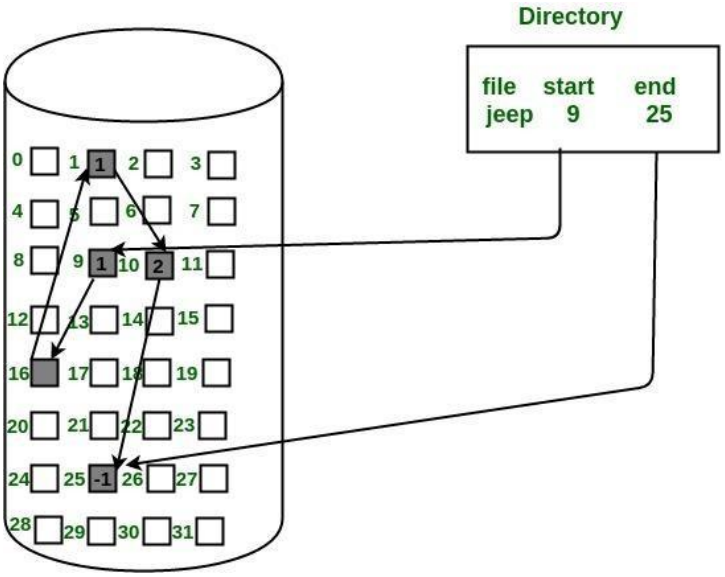
- In this figure the right hand side part is the file allocating table.
- It is consisting of a single entry for each file. It shows the file name starting blockof the file and size of the file.
- This method is best suited for sequential files.

Disadvantage:

- It is difficult to find the contiguous free blocks in the disks.
- External fragmentation occurs (i.e:- some of the free blocks may leftbetween two files).

Linked allocation:-

- In this method, every file is a linked list of disk blocks.
- It is easy to locate the files, because allocation is on an individual block basis.
- These disk blocks are present all over the disks.
- Every block contains a pointer for the next free bocks.



- These pointers are available to users.
- Ex:- there is a file “sort”, which is consist of 7 blocks. It starts from block 8and continues to block 15 and from block 15 to block 22 and so on finally itis ended with the blocks.

Advantages:

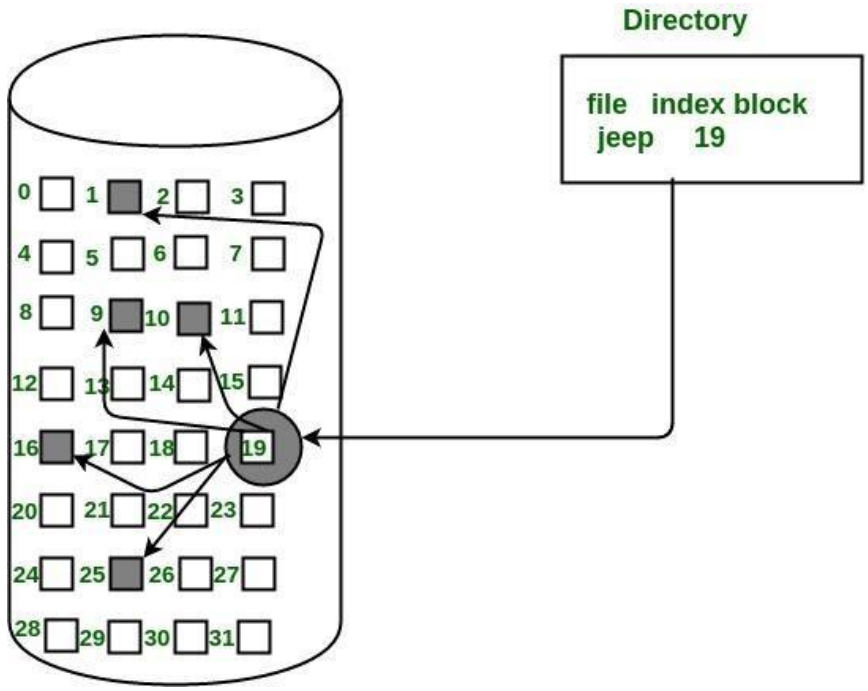
- Avoid external fragmentation.
- Suited for sequential files.

Disadvantages:

- The pointer itself occupies some memory with it in the block. So less space available for storing information.
- Takes much accessing time.

Grouped allocation or indexed allocation:-

- This method solves all the problem of the linked allocation method.
- It solves the problem by bringing all the pointers at a particular place, which isknown as index value.
- An individual block having the pointers to the other blocks.
- An individual index block is provided to every file and it contains all the disk block addresses.
- When creating a file, all the pointers are set to it.
- The fig. Shows the indexed allocation of disk space.



Advantages:

- Indexed allocation supports both sequential and direct accessing of files .
- The file indexed are not physically stored as part of file allocation table.
- When the file size increases, we can easily add some more blocks to theindex.
- No external fragmentation.

Free space management (or) disk space management:-

Generally the files are stored on disk so management of the disk space is a major problem to the designer, if user wants to allocate the space for the files we have toknow what stocks on the disk available.

- Thus we need a disk allocation table in addition to the **file** allocation table.
- To keep track of free disk space, the file system maintains a free space list. The free space list records all the disks bocks which are free i.e not allocated some other files.
- To create a file, we search the free space list. When a file is deleted its disk space is added to the free space list.
- These are the number of techniques used for disk space management:- 1:-bit sector or bit value.

  2:-chain free points or linked free space list.3:-
  index block list.

  Bit sector or bit table

  - A bit vector is a collection of bits, in which each block is represented byone bit.
  - If the block is free, the bit is 0.
  - If the bock is allocated the bit is 1.
  - Ex:- consider a disk where blocks 4,8,14,17 are free

1 1 1 0 1 1 1 0 1 1 1 1 1 0 1 1 0 1
4 8 14 17

Chain free points or linked free space list:-

- Another approach is to link all the free space blocks together keeping a pointer tothe first free block.
- This block contains a pointer to the next free disk block and so on.
- In an example we keep a pointer to the block 4, as the first free block 4 would contain a pointer to block 8, which would point to block 14, which would block to point to 17 and so on.

Indexed block list:-

- The chain free points are not very efficient to traverse the list.
- In index block list technique it would store the address of n free blocks in thefirst free block.
- The n-1 of these are actually free.
- The last one is the disk address of another block containing the address ofanother 'n' free blocks.

Advantages:

The address of large number of free of blocks can be found quickly.

File protection/sharing of files:-

Any information present in the computer system must be protected from physicaldamage and improper access.

- Files can be damaged due to h/w problems such as temperature and validationand may be deleted accidently.
- So, there is need to protect these files. There are many methods for providing protection to various files.
- File protection is depending on the system
  -in a single user system we can provide protection by simply removing floppydisks and storing them at a safe place.
  -but in multi-user system, there are various mechanism used to provideprotection. They are:-
  1-type of access2-
  access control
  3-other protection approaches (such as password).
  Type of access:-
- We can easily provide protection by prohibiting access**.**
- Controlled access is provided by protection mechanism. These mechanisms canaccept or reject an access depending on the type of access.
- The various operations that can be controlled are:-
  - Read:- helps to read from the file
  - Write:- helps to write or rewrite the file
  - Execute:- helps to execute a stored file
  - Append:- helps to write new information at the end of file.
  - Delete:- helps to delete the file
  - List:- helps to list the name and attribute of the file

- Various operations such as renaming, copying, editing of a file can be controlled.
- Different protection mechanisms are introduced for various systems and every mechanism has its own advantages and disadvantages.

Access control:-

- In this approach of protection, access depends on the identity of the user
- Every user uses a different type to access a file or directory
- The most common method to make a list with identity of each user and theiraccess control.
- When a user requests an access for file, then first it checks the access listrelated to that file.
- If that particular user is listed, then the operating system allows to access that user.
- If not, then it leads to protection violation and operating system denies the request.
- Advantage:-it can handle complex methodologies.
- Disadvantage:-the list becomes very large, when the no. Of user increases. So itis very difficult to maintain and construct a list.
- In order to solve this problem, access control is introduced. The system classifiesthe user into three different categories related to each file.
  - Owner:-it is the user who creates the file.
  - Group:- it is the set of user who shares the file and requires the same time toaccess.
  - Universe:-refers to all other user of system that constitutes a universe.

Other protection approaches (password):-

- Another protection approach is to use a password for every file.
- So accessing a file is controlled by password.

Disadvantages:

- User has to remember a large no. Of password.
- If a single password is used for all the files then if it gets discovered, then it makesall the files accessible.

Unit-7

System programming

System programming:-system programming is the activity of programming system software. The primary distinguishing characteristics of the systemprogramming when compared to that application programming aims to produce software which provides services to the user directly whereas systems programming aims to produce software and software platforms which provide services to other software.

System programming requires a greater degree of hardware awareness.

Application program:-

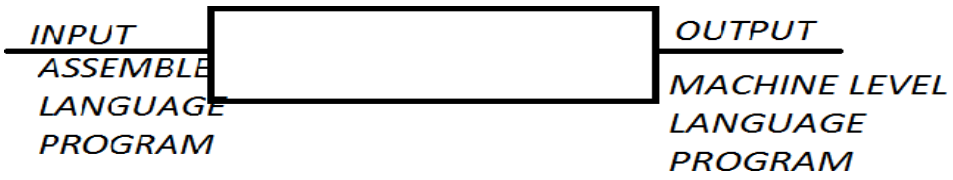
1. Application software is a set of one or more than one programs which are designed to carry out operation for a specified application.
2. For example payroll packages are designed to produce pay slip as the majorproduct. An application package for processing examination result produced mark sheet as the major product.
3. The person who prepares the application program is known as application programmer.
4. Now a days application packages are used for application such as banking, administration, insurance, publishing, manufacturing science and engineering.

System software:-

1. System software also known as system packages. These are the set of one or more than one program which are designed not to perform specific application. But these are designed to operate computer system properly.
2. The system programs helps or assist human for performing several application such as input and output data to the system.
3. It also executes the application program.
4. It manages and monitors the activities of all hardware such as memory, printer, keyboard etc.
5. These are very complex to design. So rarely it is designed in houses. These are designed by system programmers.

Assemblers:

1. A computer program which translates an assembly language program toits machine language equivalent is known as assembler.
2. The assembler is a system program which is supplied by the manufacture.
3. A symbolic program written by a programmer is assembly language is called a source program. After the source program has been converted into machine language by an assembler it is referred to as an object program.

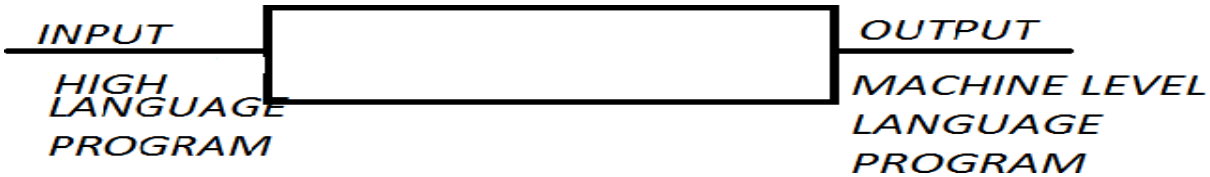


4. The input of the assembler is the assembly language program and outputfrom the assembler is the machine language program.

Complier:

1.A complier is a program that translates the high level language intomachine level language by reading the entire source code.

2-A program written by a programmer in high level language is called source program that has been converted into machine language by acomplier is referred to as object program.



3. So input to a complier is known as source program and output from acomplier is known as object program.

4. A single complier cannot translate all the high level language intomachine level language should have a dedicated complier for itscompilation.

5. Complier is a large program which resides in secondary. When it isrequired it is copied into main memory.

Interpreter

An interpreter is also a translator that translates the high level language intomachine level language by reading one by one.

Here translation and execution alternate for each statement encountered in highlevel language program.

An interpreter translates the instruction and the control unit executes theresulting machine code so on.

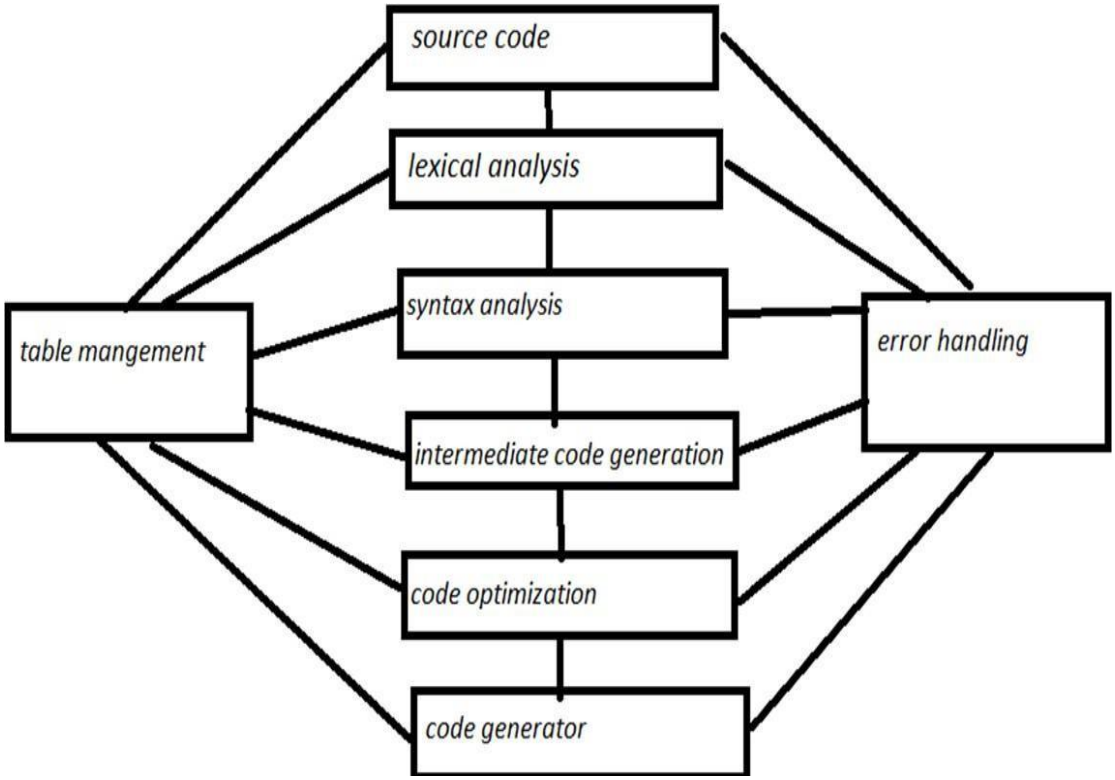
It is simple to write and required less space in main memory for storage.As one by

one line is translated so it is slower.

Complier	Interpreter
1.Complier is software that translates the high level language into machine language by reading the entire code at a time.	1.Interpreter is a software thattranslates the high level language into machine language by reading one statement at a time.
2. Repeated compilation is not necessary for repeated execution of a program.	2. Repeated interpretation is necessary.
3. slow response to the change in source code.	3. Fast response to the change in source code.
4. it is a complex program as compared to interpreter.	4. it is a simple program.
5. it requires large memory space in computer.	5. easy to write and do not require large memory space in computer.
6. it is faster.	6. it is slower.

Stages of complier

The complier takes an input a source program and produces as output an equivalent sequence of machine instruction. The complier does this transition by some sequence of gates or phase.



Lexical analyzer/scanner:-

- Lexical analysis is the first phase of compiler which is also termed as scanning.

- Source program is scanned to read the stream of characters and those characters are grouped to form a sequence called lexemes which produces token as output.

Token: Token is a sequence of characters that represent lexical unit, which matches with the pattern, such as keywords, operators, identifiers etc. This separators character of sources language into groups that logically belong together, these groups are called tokens. The usual tokens are keyword operator symbol.

Syntax analyzer/parser:- The output of the lexical analyzer is passed to this syntax analyzer. The syntax analyzer check whether the statement is valid or not every language has its production.

If sentence follows this rules this rules then the sentence is valid.

To check the validation of a sentence two techniques are used:-

- Top down approach
- Bottom up approach

Intermediate code generation:- This phase uses the structure produces by the syntax analyzer to create a stream of simple instruction. These instructions are similar to assembly language.

Code optimization:-This is an optional phase, whose job is to improve the intermediate code. So that the ultimate object program can run faster.

Code generation:-This phase produces the object code by deciding where the memory space will be allocated to the variables, literals and constant.

Table management:- This portion of the complier keeps track of the names used by the program and records essential information. The data structure used to record this information is called a symbol table.

Error handler:-The error handler is involved when an error in the source programis detected. Generally the error occurs at the syntax analyzer phase.

Both the table management and error handler routines interact with all the phase of the complier.

References

1. “Operating System Concepts” by Avi Silberschatz ,Peter Baer Galvin and Greg Gagne
2. “Operating System” by Er. Rajiv Chopra
3. “Operating System and System Programming” by P. Balkrishna Prasad
4. “Operating Systems” by Vijay Shukla
5. “Operating System” by Stuart Madnick and John Donovan.
6. https://www.geeksforgeeks.org
7. https://nptel.ac.in
8. https://en.wikipedia.org