PNS SCHOOL OF ENGINEERING & TECHNOLOGY
Nishamani Vihar, Marshaghai, Kendrapara

LECTURE NOTES
ON
OPERATING SYSTEM

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

4^{TH} SEMESTER

PREPARED BY
MR. BISWARANJAN SWAIN
Lecturer, Department of Computer Science & Engineering

SYLLABUS

1. INTRODUCTION

1.1 Objectives and Explain functions of operating system.
1.2 Evolution of Operating system
1.3 Structure of operating system.

2. PROCESS MANAGEMENT

2.1 Process concept, process control, interacting processes, inter process messages.
2.2 Implementation issues of Processes.
2.3 Process scheduling, job scheduling.
2.4 Process synchronization, semaphore.
2.5 Principle of concurrency, types of scheduling.

3. MEMORY MANAGEMENT

3.1 Memory allocation Techniques
- Contiguous memory allocation
- non contiguous memory allocation

3.2 Swapping
3.3 Paging
3.4 Segmentation, virtual memory using paging,
Demand paging, page fault handling.

4. DEVICE MANAGEMENT

4.1 Techniques for Device Management
- Dedicated,
- shared and
- virtual.

4.2 Device allocation considerations I/O traffic control & I/O Schedule, I/O Device handlers.
4.3 SPOOLING.

5. DEAD LOCKS

5.1 Concept of deadlock.
5.2 System Model
5.3 Dead Lock Detection
5.4 Resources allocation Graph
5.5 Methods of Deadlock handling
5.6 Recovery &Prevention, Explain Bankers Algorithm & Safety Algorithm

6. FILE MANAGEMENT

6.1 File organization, Directory & file structure, sharing of files

6.2 File access methods, file systems, reliability

6.3 Allocation of disk space

6.4 File protection, secondary storage management.

7. SYSTEM PROGRAMMING

7.1 Concept of system programming and show difference from Application Complier:

7.2 Compiler , functions of compiler.

7.3 Compare compiler and interpreter.

7.4 Seven phases of compiler, brief description of each phase.

UNIT-1

INTRODUCTION

INTRODUCTION:

System is nothing but environment. Environment consists of components.

Operating system:-

⇒ The basic objective of the operating system is to operate several components associated with computer system.

⇒ System is an environment consists of several components.

⇒ System having two components (basically)

i) Hardware components

ii) Software components.

i) Hardware components:-

⇒ There are physical components.

⇒ The components which are visible, touchable part of the computer system that can perform basic function is known as hardware.

Eg. Input devices, O/P devices, m/m, Processor.

ii) Software components:-

⇒ These are the untouchable components having logical existence.

⇒ These are the set of programs that controls the operation of the physical components.

⇒ Programs are the set of instructions.

Software can be of two types:

System S/W & Application S/W

System software:-

⇒ It is meant for running of a system.

⇒ It is a set of programs designed to make function able of different components.

Application software:-

⇒ These are the S/W related to user's requirement.

⇒ Thus application software can't be executed alone. It always take the help of system software to be executed.

⇒ These are the set of programs that are designed to execute diff. app. Program. e.g. word processor, spread sheet, railway/ air reservation.

Operating System:

Operating system is a system software that acts as an intermediary between the user of acomputer 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".

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 anefficient 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.

FUNCTIONS OF OPERATING SYSTEM

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

As Command Interpreter:

Generally the CPU cannot understand the commands given by user. It is the functionof 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 Operating System interacts with the computer hardware.

As 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 be shared 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 afixed 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 usersat the same time.

Memory Management:

It keeps track of the resources (memory), what part of memory is in use and bywhom, 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

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 isexecuting.

The operating system is responsible for the following functions related to process management.
Process creation (loading the prog. From secondary storage to mainmemory) Process scheduling
Provide mechanism for process synchronization Provide mechanism for deadlock handling
Process termination

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 like keyboard, mouse, monitor, disk, FDD, CD-ROM, printeretc.
Allocation and De allocation of resources to initiate I/O operation.

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
Creation and deletion of files
Creation and deletion of directories
Manipulation of files and directories
Mapping files onto secondary storage

Secondary storage Management:

It is a larger memory used to store huge amount of data. Its capacity is muchlarger 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 the secondary storage management.
The various activities are:
Free space management
Storage allocation (allocation of storage space when new files have tobe written).
Disk scheduling (scheduling the request for memory access)

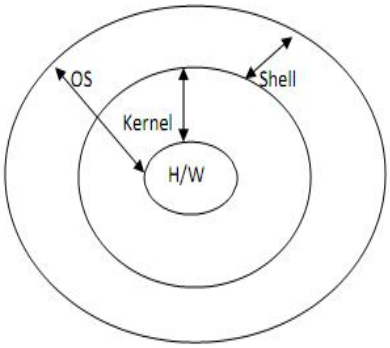
Protection/Security Management:

If a computer system has a multiple processor, then the various processes must be protected 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.

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.

COMPONENTS OF OS:-



Basically OS divided into two components, i.e.

i) Kernel

ii) Shell

i) Kernel:

⇒ It is the core part of the OS.

⇒ This part of OS deals with h/w (hardware instructions).

⇒ It is that part of OS who is always in a running mode.

ii) Shell:-

⇒ It is that part of OS who is directly related to the user.

⇒ Basically it deals with a high level language or commands or instruction.

⇒ It is also acting as a command interpreter.

Relationship between shell and kernel:-

⇒ A shell takes the instruction from the user through a high level language.

⇒ Then it converts the high level language to a machine level language (0,1 form) through an interpreter.

⇒ After getting instruction from the shell, kernel instructs the appropriate hardware to be executed.

TYPES OF OPERATING SYSTEM

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

OPERATING SYSTEM are classified into different categories according to their different features. The following 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

Advantage:

The CPU has to handle only one application program at a time so that process managementis 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 computer at 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 withsimilar needs executed simultaneously.

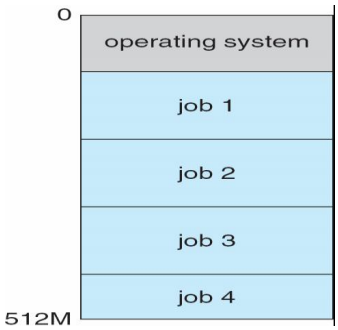
Advantage:

It is simple to implement.

Disadvantage:

Lack of interaction between user and the program.

Multiprogramming Operating System:



In a multiprogramming operating System several usercan execute Multiple jobs by using a single CPU at thesame time.

The operating System keeps several program or job inthe main memory.

When a job is submitted to the system in a magneticdisk 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 the operating 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 of I/O operation of that job.

When the I/O operation of that job is completed then the CPU switches to the next job after 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 amongthemselves.

The switching process is very fast so that the user can directly interact with the systemduring the execution of the program.
This system stores multiple jobs in the main memory and CPU execute all the jobs ina sequence.
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 next process.
The switching process is very fast, so it seems that several processes are executed simultaneously.

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.

Real time Operating System:

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

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- speedbuses 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 forexchange 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 differentnodes.
It enable different users to communicate with each other using email.

UNIT-2

PROCESS MANAGEMENT

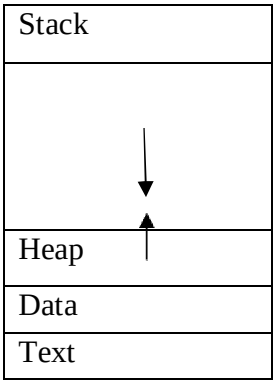
PROCESS:

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 machinecode.	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 machinelevel 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.
Stack
Heap
Data
Text
Stack section contains local variable
Data section contains global variable
Text section contains code or instruction.
Heap section contains memory which will be dynamically allocated during runtime.



PROCESS STATE:

When a process is executed, it changes its state. The current activity of that process is known as Process 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.

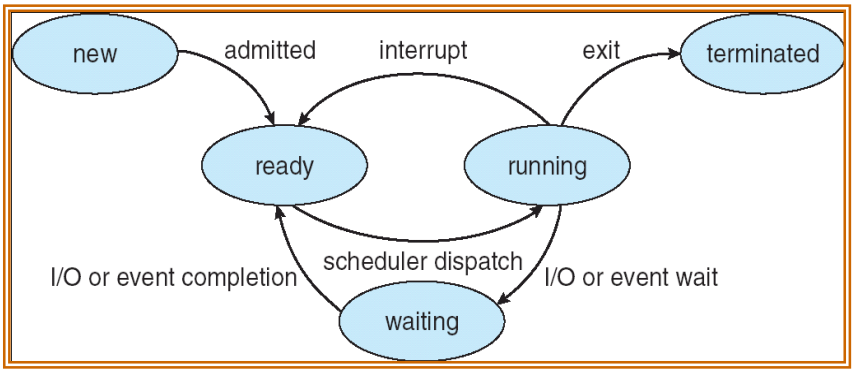


Diagram of process state

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/ blocked state:-

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 theprocess moved into a waiting state.

If a higher priority process arrives at the CPU during the execution of anongoing process, then the processor switches to the new process and currentprocess enter into the waiting state.

Terminated state:-

After completion of execution the process moves into the terminated state byexiting 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:

To represent a process the operating System needs to group all the information of a process inside a data structure. 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 a fundamental unit for Resource Allocation. Following resources could beallocated to a process.

The information stored inside the PCB includes

Pointer-It stores the starting address of the process.

Process State- This field stores or represent the current state of the process whether it is in ready/running/new/waiting/terminating.

Process ID/Number-Each process has unique ID or serial no. Each process is shown an unique no. known as its Process ID or Process Number.

Program Counter- It stores the address of the next instruction to be executed.

Register-This field contains the name of the registers which are currently used by the processor.

Scheduling Information-This field stores the information about the scheduling algorithm used by operating System for scheduling that process.

Account Information-This field contains the total no. processes, timeslice period it used.

File Management Information- It stores various information about the files used by the process.

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

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:

Scheduling Queue

Scheduler

Context Switching

Scheduling Queue

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

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.

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.

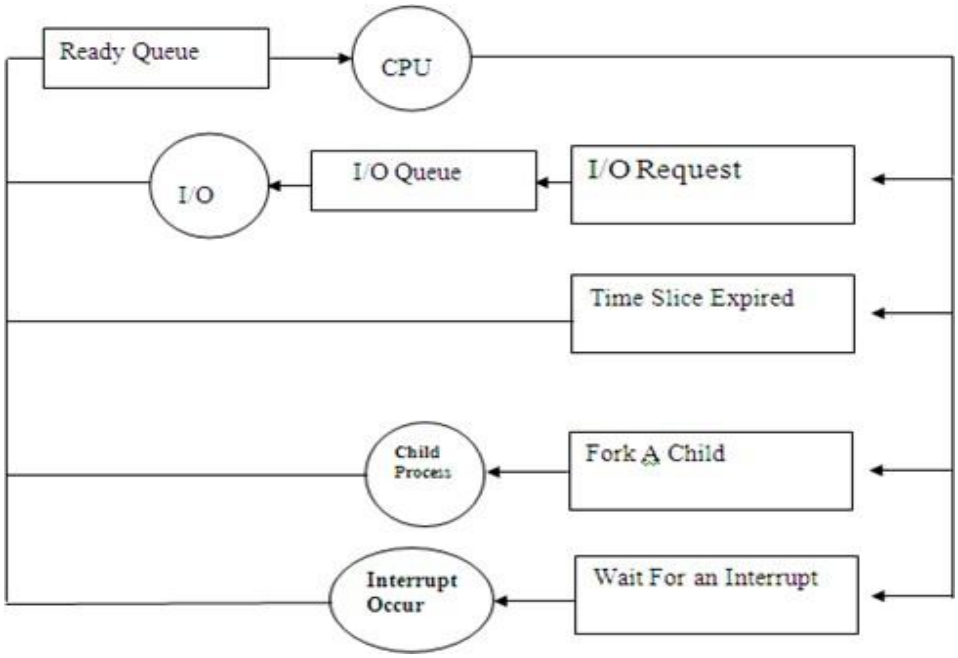
Device Queue:

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.

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 sub-process 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.



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.

Long Term Scheduler

Short Term Scheduler

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.

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/Orequest.

Medium Term Scheduler(MTS):

When a process moves from running state to waiting state and from waiting state to ready state, the transition of process occurs through a component named as middle term scheduler.

Job scheduler (long-term scheduler)	CPU Scheduler (Short term scheduler)
⇒ It is used to copy the jobs from job pool to load them into the main memory from execution. ⇒ It is otherwise known as long term scheduler. ⇒ It works in larger interval i.e. executed after a set of jobs are completed.	⇒ It copies the job from main memory to CPU for execution. ⇒ It is otherwise known as short term scheduler. ⇒ It works in smaller intervals. Here the intervals equals to the execution of a single job.

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.

SCHEDULING ALGORITHM

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

First Come First Serve scheduling(FCFS)

Shortest Job First(SJF)

Priority scheduling
Round Robin Scheduling
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.

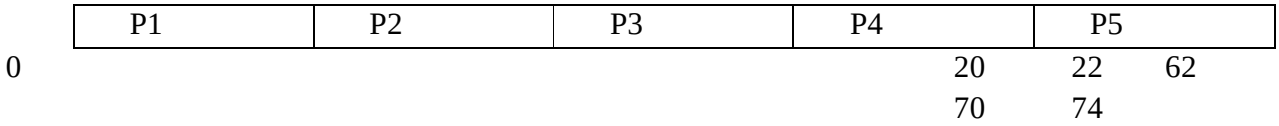
Let the process arrives in the order p1, p2, P3, p4,p5.

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:



Waiting time:

P1=0

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

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

Turn Around Time(TAT):

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

Shortest Job First Scheduling(SJF)

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 executing process.

A preemptive SJF algorithm will pre-empt the currently executing process where as a non preemptive SJF algorithm will allow the currently running process to finish its CPU burst.

Preemptive SJF scheduling is sometimes called “shortest remaining time firstscheduling”. 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=2

0-

4=16

P3=1

5-6=9

P4=30-8=22 A.W.T=0+16+9+22=47/4=11.75**T.A.T**
P1=15- 0=15P2=30-4=26 P3=20-6=14 P4=38-8=30
A.T.A.T=15+26+14+30=85/4=21.25

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 theready queue.

In case of FCFS scheduling the ready queue is a FIFO queue. But in RR scheduling the ready queue is a circular queue.

Round Robin Scheduling is a purely preemptive scheduling algorithm. Because after every timeslice 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)=30P2=30-15=15
P3=(35-20)+(50-40)=20P4=(20-15)+(40-25)=20
A.W.T=(30+15+20+20)/4=21.25

T.A.T

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

Inter-process 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: 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.

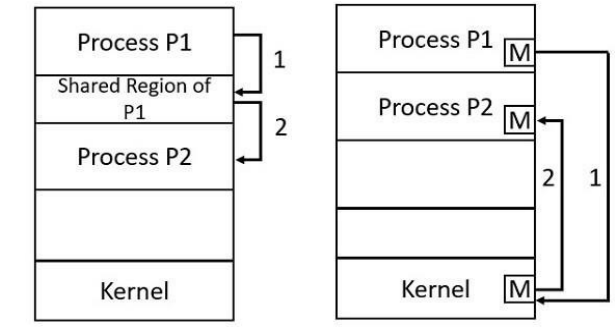
Co-operating process:-

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.

Ways to Implement IPC

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.

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.



PROCESS SYNCHRONIZATION:

It is the task of coordinating the execution of processes in a way that no two processes can have access to the same shared data and resources.

It is needed in case of multiprocessing system. This may create inconsistency in data. To avoid this problem the processes need to be synchronized.

Sections of a Program:

There are four essential elements of the process.

Entry Section:

It is part of the process which decides the entry of a particular process.

Critical Section:

This part allows one process to enter and modify the shared variable.

Exit Section:

Exit section allows the other process that are waiting in the entry section to enter into the critical section. The process that finished int execution should be removed through this section.

Remainder Section:

All other parts of the code, which is not critical, entry or exit are known as remainder section.

Critical Section Problem:

When one process is executing in its CS, no other process is to be allowed to execute in its CS i.e no two process can execute in their CS at the same time.

```
do
{
| Entry |

        Critical section

| Exit Section |

        Remainder section
} while(true)
```

There are different methods to solve critical section problem such as synchronization h/w, mutex locks, semaphores.

SEMAPHORE

⇒ It is a synchronization tool, denoted as 'S' which is an integer variable whose value can be changed and altered.

⇒ Its value indicates the status of shared resources, a process which needs the resource, will check the semaphore for determining the status for the resource (available/unavailable).

⇒The value of the semaphore variable can be changed by two operations.

i) Wait (P)

ii) Signal (v)

Wait(S) :-

⇒ The wait operation decrements the value of its argument S, if it is positive.

If S is negative or zero, then no operation is performed.

```
if (S>0)
  wait(s)
  {
        while (S<=0);
         S--;
   }
```

Signal (S) :-

⇒ The signal operation increments the value of its argument S

```
Signal(s)
{
      S++;
}
```

⇒ When one process modifies the value of the S, no other process can simultaneously modify the same S value.

Types of Semaphore:

There are two types of semaphore:

i) Binary Semaphore

ii) Counting Semaphore

i) Binary Semaphore:-

⇒ Binary semaphore can take 2 values i.e 0 or 1.

⇒ Initially the value of S is set to 1, and if some process wants to use some resource then the wait() function is called and value is set from 1 to 0.

The process then uses the resource and when it releases the resource signal() function is called .

The value of S becomes 0 to 1.

When the value of S=0 then other processes wait.

⇒ The lock used by the binary S is termed as MUTEX lock.

```
do
{
Waiting (mutex);
// c.s
Signal (mutex);
//reminder section
{while (true);
```

ii) Counting Semaphore -

⇒ The counting semaphore is applicable for multiple instances of resource type.

⇒ Each process that wants to use the resource performs wait operation on the S.

⇒ When a process release the resource, it perform signal operation.

⇒ When the count of the S goes to zero all the resources are being used.

Questions:-

1. Draw the states of process.
2. Draw the PCB.
3. What are the different types of scheduler?

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:-

When a process resides in memory to give the protection it will take the help of two registers.

i) Base register

ii) Limit register

Base register holds the smallest physical memory address (the memory location from where the process starts).

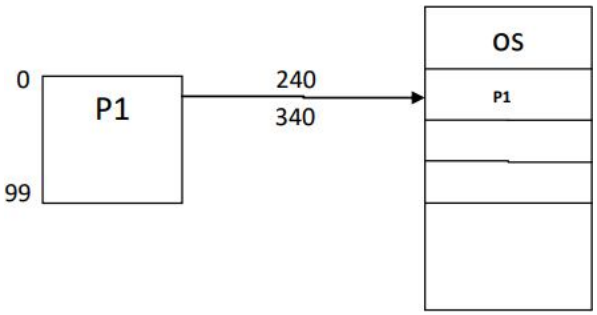
The limit register holds the size of the range.

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 logicaladdress 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 memory management unit (MMU).

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.

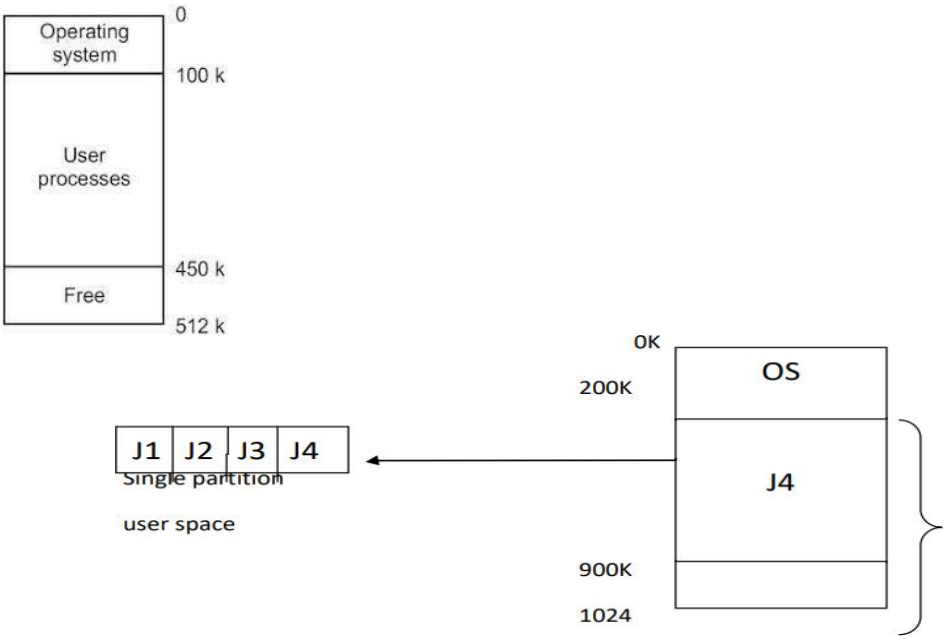
- Generally, there are two methods used for partitioning the memory allocation.
  - *Contiguous memory allocation*
  - *Non-Contiguous memory allocation*

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 resides in the low memory.



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 oneprocess at a time, because the user space treated as a single partition.

Advantage:-

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

Disadvantage:-

Memory is not utilized property.
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 andrest of main memory 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 "InternalFragmentation" 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 causes internal 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.

Fixed variable partition:- (unequal size partition)

o In this scheme the user space of main memory is divided into number of partitions,but the partitions sizes are different length.

o The operating system keep a table which indicates, which partition of memory are available and which are occurred. This table is known as "Partition Description Table"(PDT).

o When a process arrives and needs allocation or memory, we search for partition which is big enough 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 toimplement.

Disadvantage:-Suffers from internal as well as external fragmentation.

Figure 4: Fixed Partitioning

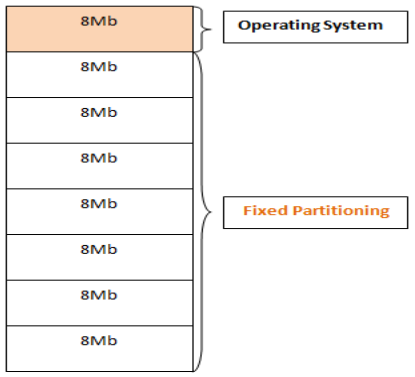
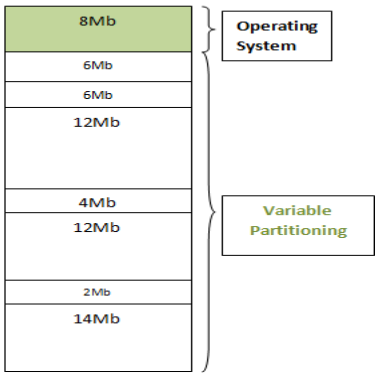


Figure 5: Variable Partitioning



(Variable partition)

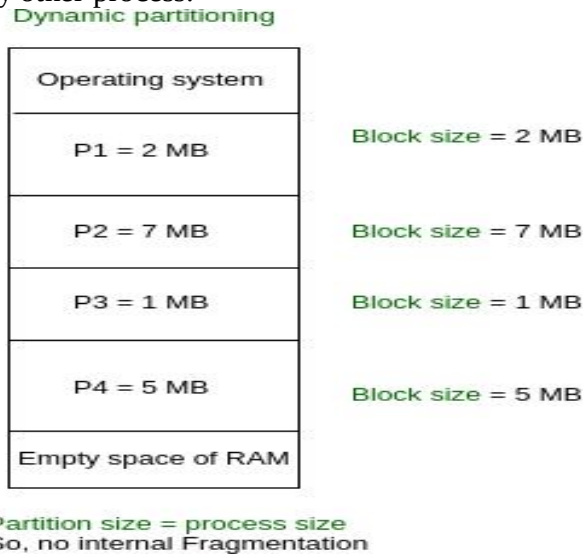
Dynamic Multiple Partition :-

o To overcome/eliminate some of the problems with fixed partition, another method isdeveloped known as "Dynamic Partitioning".

o In this technique, the amount of memory allocated is exactly the amount of memory a process

requires.

- In this method the partition are made dynamically.
- Initially when there is no process in the memory, the whole memory is available forallocation and 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 thatprocess is allocated.
- 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 canbe used by other process.



Advantage:-

i. Partition changed dynamically. So no internal fragmentation.
ii. Efficient memory and CPUutilization.

Disadvantage:-

i. Suffers from external fragmentation.

Partition Selection Algorithms:-

Whenever a process arrives and there are various holes large enough to accommodate it, the operating system may use 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:-

This algorithm is fast because very little search isinvolved.

Disadvantage:-

Memory loss may be high.

Best fit:- In this algorithm the operating system scans the free storage list and allocatethe smallest partition that is big enough for the process.

Advantage:-

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 largest partition to the process.

Disadvantage:-
Maximum interval fragmentation.

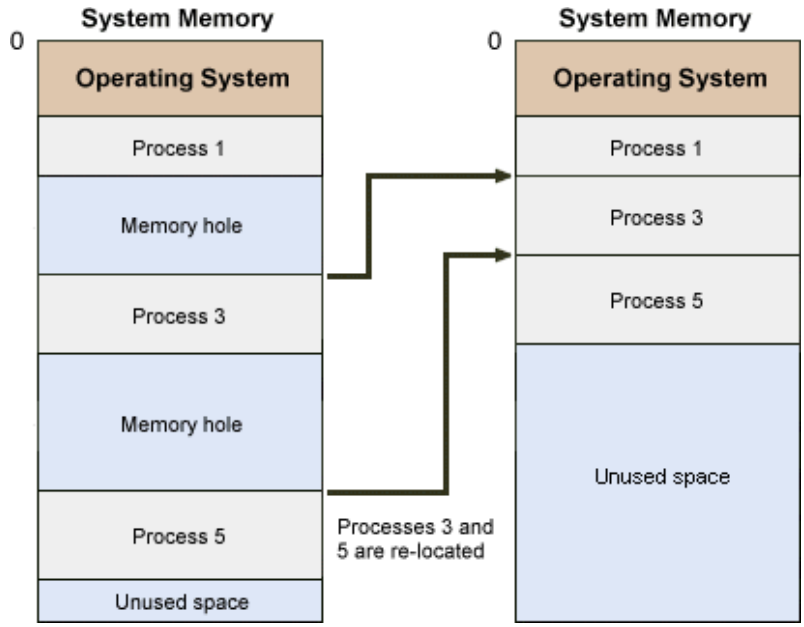
Compaction:-

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 used by another program.

Let the size of the freed memory be S, the next program to be run on the memory may need a space 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 remains unutilized.

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 orportion remains 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 allocationis known as Non-contiguous allocation.

The two popular schemes in Non-contiguous allocation are paging & segmentation.

SWAPPING:-

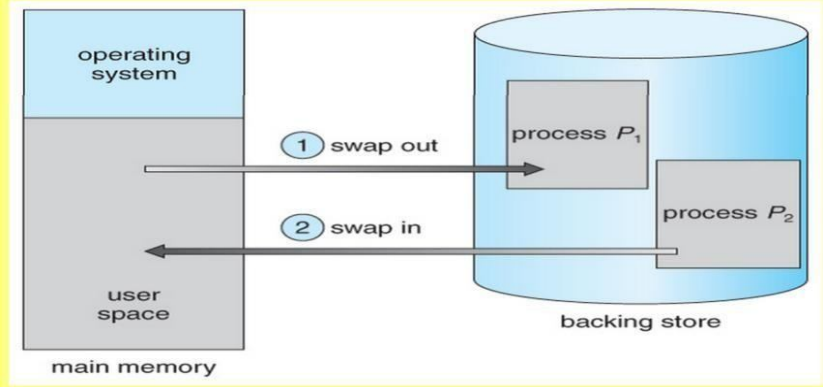
- Swapping is the method to improve main memory utilization.
- When a process is executed it must be in the main memory.
- A process can be swapped out temporarily to secondary memory or hard disk or backing memory and then again brought back to secondary memory for execution. This techniqueis known 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 toprimary memory.



Paging

Paging is an efficient memory management scheme because it is Non-contiguous memoryallocation 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.

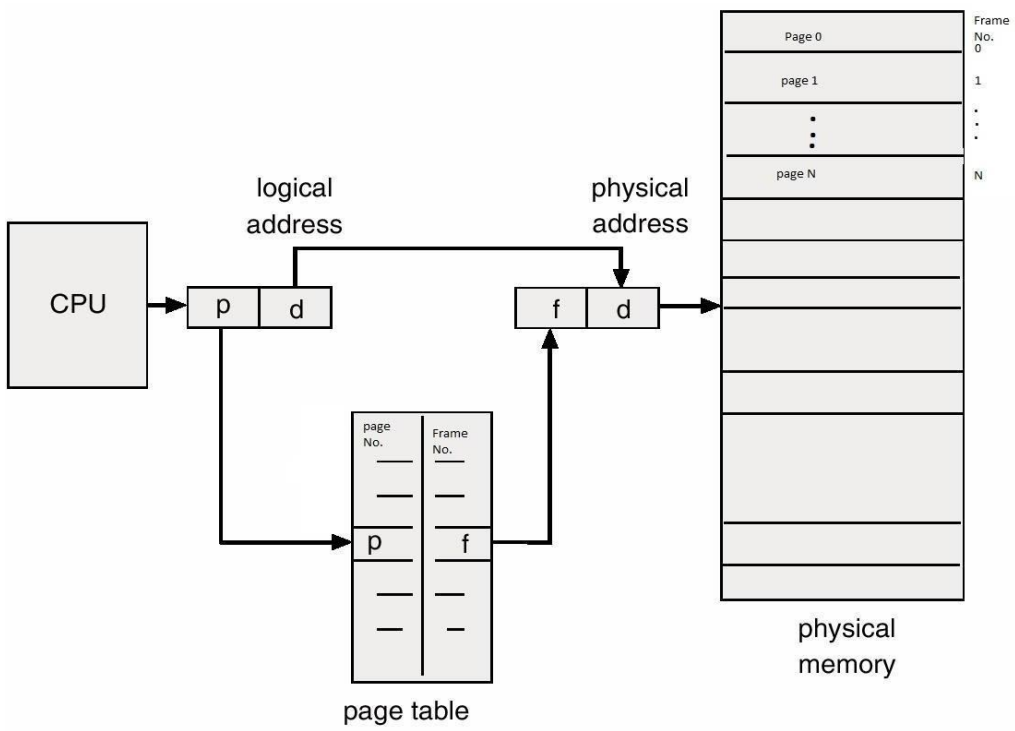
The basic idea of paging is physical memory/ main memory is divided into fixed sizeblocks called as frames.

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 available memory 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 isgenerally a power of 2, such as 4MB, 8MB, 16MB etc

The page map table specifies which page is loaded in which frame, but displacement or offset is common.

The paging has no external fragmentation, but there may be internal fragmentation .Inpaging it is called as **page break**.

Advantage

It supports time sharing system

It does not effect from fragmentationIt support virtual memory.

Disadvantage

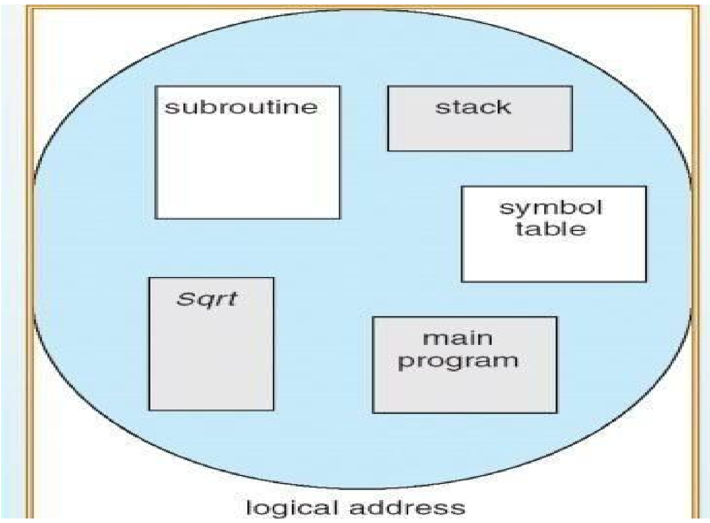
The scheme may suffer “page break”.

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 some containing data.
But he view the memory as a collection of variable sized segments and there is no ordering of segments.

Segment is a memory management technique that supports **user’s view of memory.**



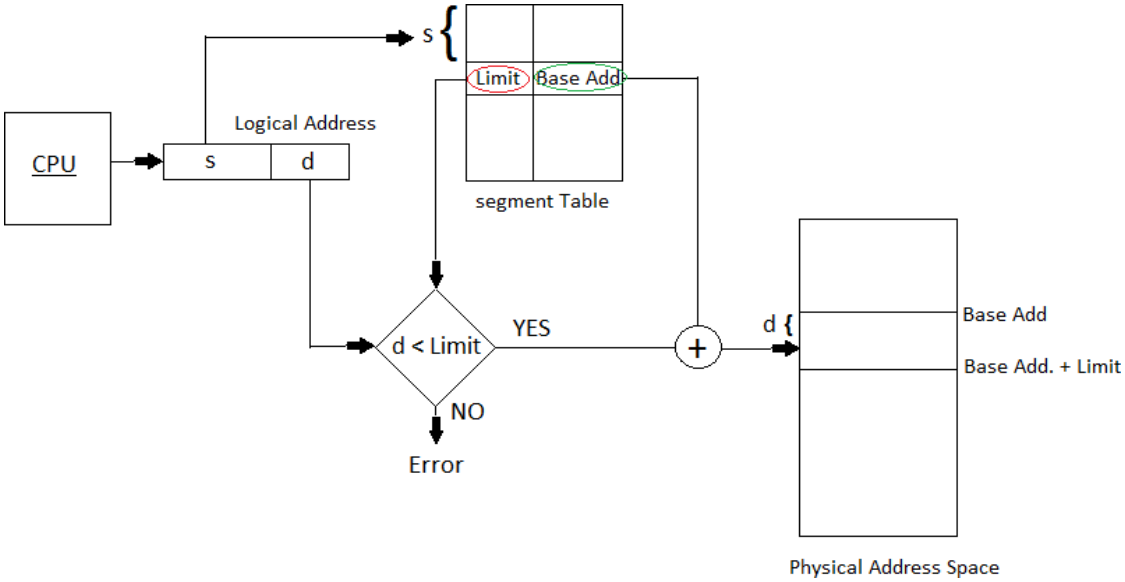
A segment can be defined as a logical grouping of instructions, such as subroutine, array or a data area.
Every program is a collection of these segments. Here the logical address is a collection of segments.
Each segment has a name and length.
Segmentation is a technique for managing these segments.
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 offset value is 100K.the operating system searches the entire main memory for freespace 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)

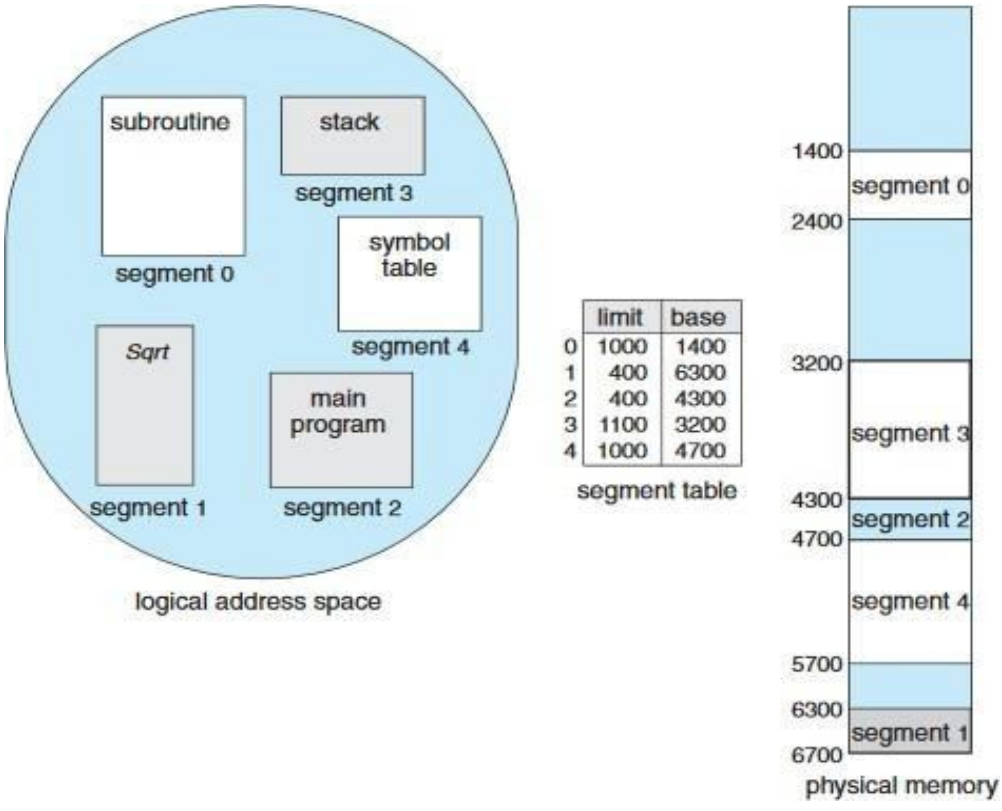
The segment number is used as an index into the segment table.
The offset is compared with segment limit.

Offset should be less than or equal to limit, elsethere 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 inmain memory then an interrupt occurs to the operating system called as **page fault.**

When the page fault happens, 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 isvalid 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 desired 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 execution of the process starts where it was left.

Paging vs Segmentation:

Paging	Segmentation
The main memory partitioned into frames or blocks	The main memory partitioned intosegments
The logical address space divided into pagesby compiler or memory management unit.	The logical address space is divided into segments specified by the programmer.
It may suffer from page break or internal 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 offset to 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 to secondary 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 occupyvery less space on actual physical memory

Demand Paging:-

Demand paging is the application of virtual memory. It is the combination of paging and swapping.

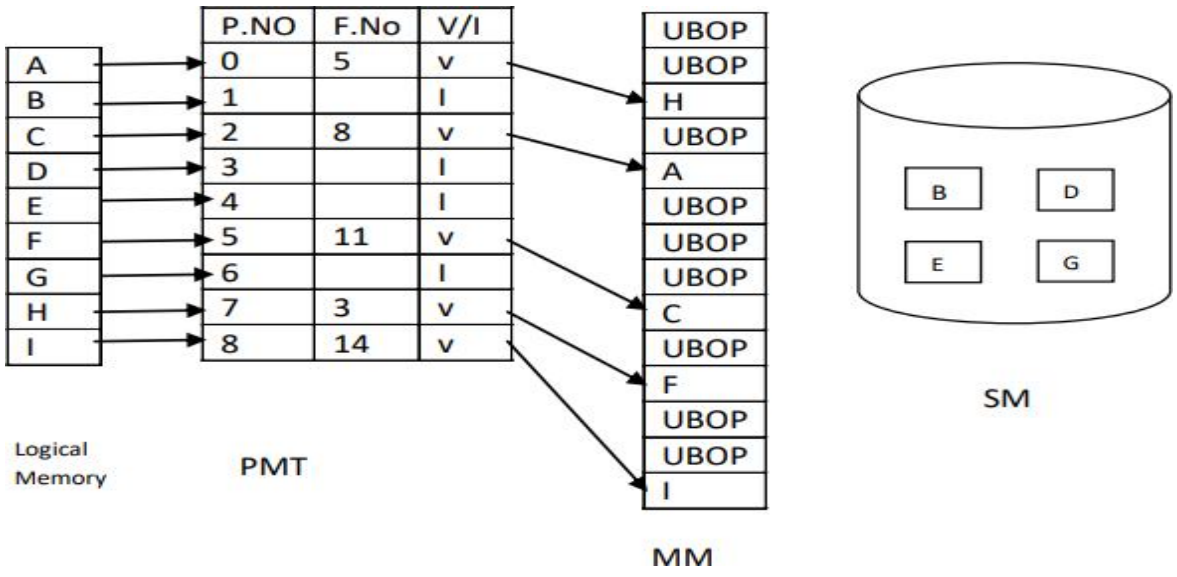
The criteria of this scheme is "a page is not loaded into the main memory from secondary memory, until it is needed".

So, a page is loaded into the main memory by demand, so this scheme is called as "DemandPaging".

For ex: Assume that the logical address space is 72 KB. The page and frame size is 8KB. Sothe logical 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 are loaded in the secondary storage.

Whenever those pages are required, the operating system swaps-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 areused by other process(UBOP).

QUESTIONS

1) Define segmentation with one example ?
2) Define multiple partitioning.
3) What is swapping ? Explain the principle of swapping with diagram.
4) Define swapping ? Explain the role of swapping in virtual memory management.
5) What is page fault ? Explain the page fault handling techniques.
6) Define page. Explain memory management technique using paging .
7) Explain demand paging method of memory management.
8) Define virtual memory. Explain virtual memory using segmentation method of memory management.

UNIT-4

DEVICE MANAGEMENT

Device management technique:-

Device Management in Operating System manages device communication via their respective drivers.

It does the following activities for device management:

Keeps tracks of all devices. The program responsible for this task is known as the I/O controller.

Decides which process gets the device when and for how much time.

Allocates the device in the most efficient way.

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 releases the deviceonce 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 differentprocesses' requests.

All conflicts for device need to be resolved but predetermined policies to decide which request ishandled 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,ensuringthat 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 issatisfied, first. Then, the process of scheduling, is applied here, and it is known as I/O scheduler.

I/O device handler:-

I/O processes the I/O interrupts, handles error condition, and provides detailed scheduling algorithms, which are extremely device dependent.

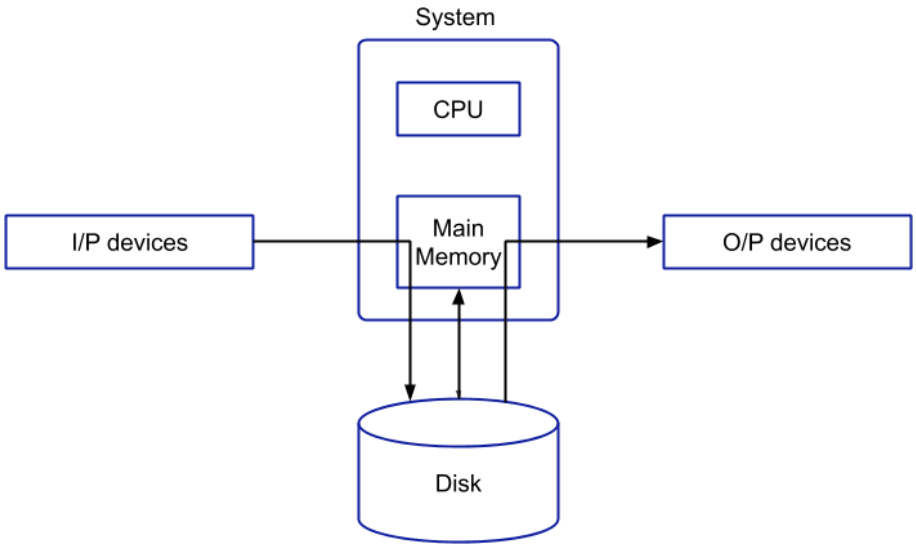
Each type of I/O device has own device handler algorithm like FCFS, SSTF, SCAN.

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 be used 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 aprinting schedule.

Race condition:-

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

QUESTIONS:

1) Define SPOOLING .
2) Define device management ? Explain the techniques of device management.

UNIT-5

DEAD LOCKS

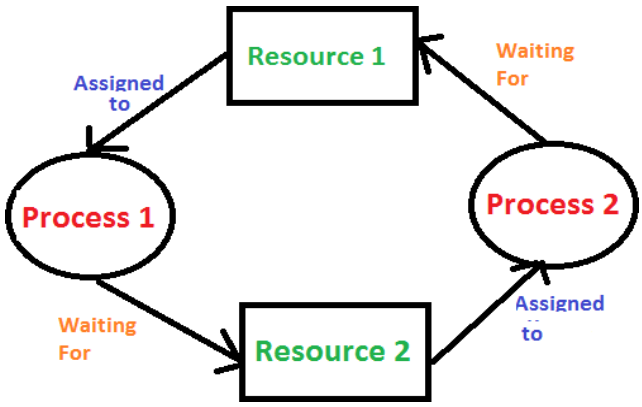
System Model:

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 utilize 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 bereleased.
   Example: close( ), free( ), delete( ), and release( ).

Deadlock: *Deadlock* is a situation where a set of processes are blocked because each process is holding a resource 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 is acquired 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 resource must be held in a non-sharable mode. That means only one process can use that resource at a time.

So that if the resource is not free then the requesting process has to wait till the resource is released by theother process.

2) HOLD AND WAIT:-There must be a process which is already holding using one resource and requesting (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 be 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 waiting processes.

P0 is waiting for the resource held by P1.

P1 is waiting for the resource held by P2. P2 is waiting for the resource held by P3.

_ _

_ Pn-1 is waiting for the resource held by P0.

Resource allocation graph (RAG):-

- A diagrammatic representation of resource distribution. 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.

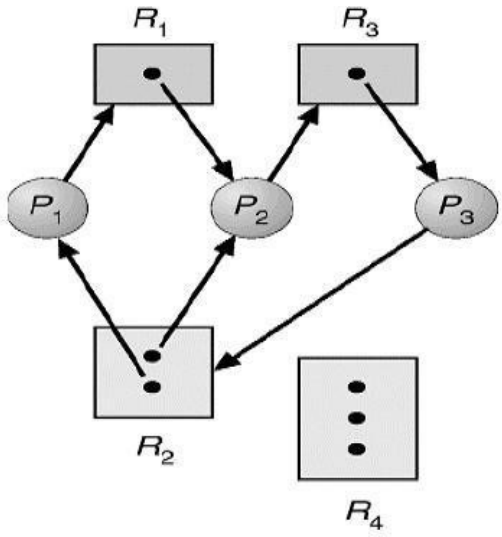
ASSIGNMENT EDGE:- Whenever a resource is allocated to a process the request edge is converted toan assignment edge from the instance of the resource to the process.

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 instances per resource then a cycle indicates that "there may bedeadlock".

The resource allocation graph shown in figure has thefollowing 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.

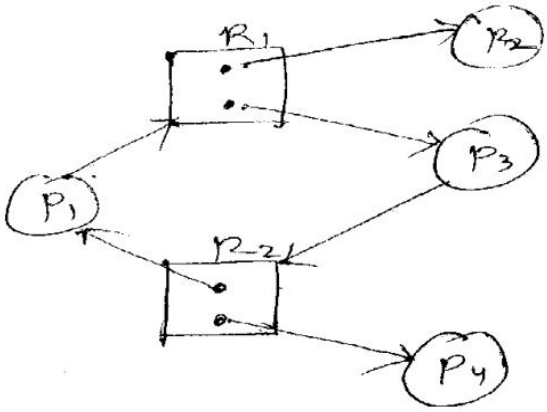
The process states are:

- Process P1 is holding an instance of R2 and waitingfor an instance of R1.
- Process P2 is holding an instance of R1 and R2 and waiting
- 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

Resource allocation graph with a cycle but no deadlock:-



Methods for Handling Deadlocks

Deadlocks can be handled by one 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 R5 such request will not be granted, only request for resources more than R5 will be granted.

DEADLOCK AVOIDANCE

⇒ Deadlock can be avoided by considering the future request by the process.
⇒ For the deadlock avoidance 2 approaches should be followed.
1) Never start a process if it demands tends to deadlock.
2) Don't allocate any additional resource if this allocation lead to deadlock.

SAFE STATE: -

⇒ A state is said to be safe, it system can allocate the resources to each processes by following some sequence/ order without causing deadlock in the system.

⇒ A system is said to be safe if there only exist a safe sequence (fig-1)

⇒ Safe state = safe sequence = no deadlock

⇒ Unsafe state = unsafe sequence = deadlock

⇒ There are 2 deadlock avoidance algorithm.

i) Resource request algorithm

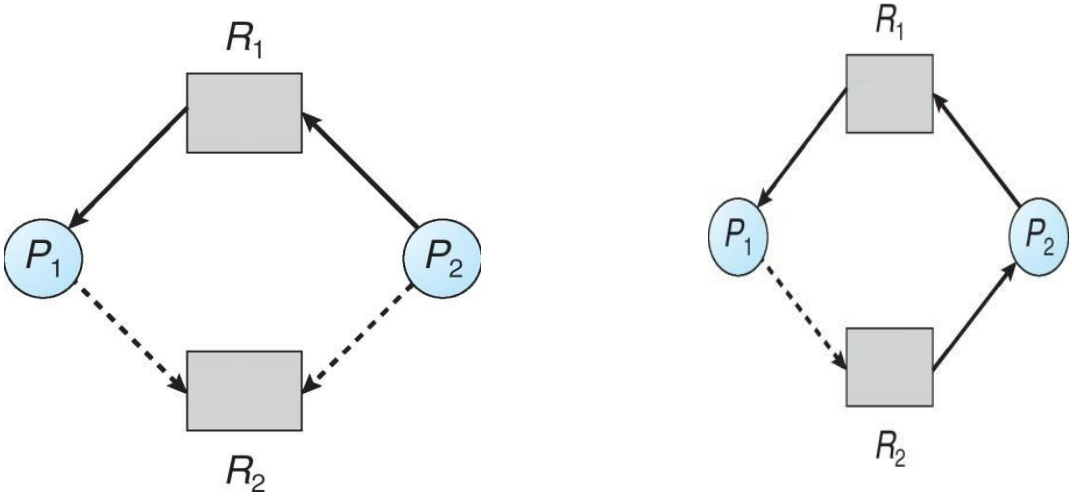
ii) Banker's algorithm.

Resource request Algorithm :-(Multiple Resources Having single instance)

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

In this graph, beside assignment edge and request edge, third edge known as "claim edge" is added. 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 claim edges for the same resource then it can avoid. If only one of the processes is allocated the resource R1 then a deadlock can arise.

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 + request(i)
   need (i) = need (i) - request (i)

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

The resource allocation algorithm 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.
- 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 that process.

- 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 to determine whether or not a request generated by a process for a resource, would lead the system to an unsafe state.

The algorithm uses several data structures such as vector & matrices

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 a 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 defines the number of resources of each type currently allocated to each process.

Ex. If allocation [i,j]=k, then process Pi is currently allocated k instancesof 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 follows:

⇒ Work – it is the vector of length m.

⇒ Finish – vector of length n. initialization work = available.

Finish [i] = false for I = 0 to n – 1.

⇒ For find an I, s.t. both. Finish [i] == false and need(i) <= 1 work.

If no such I exist, go to step 4.

⇒ Work = work + allocation I .

finish [i] = true, go to step-2.

⇒ Finish[i] == true, when the system is in safe state.

Steps to solve banker's algorithm:-

Steps-1:-calculate the need matrix. Need = max – allocation.

Step-2:- calculate the order of execution of processes (safe sequence).

i) Check need ≤ available, if no. then the process must wait, otherwise the process will execute and after execution it should release the resources.

ii) Update the available, available = available +allocation

Step-3:-continue the process (1st iteration) to get a sequence.

Step-4:- if any process doesn't get a chance to execute in the 1st interaction, then continue to make them execute.

Process	Max			Allocation			Available		
	A	B	C	A	B	C	A	B	C
P0	7	5	3	0	1	0	[3	3	2]
P1	3	2	2	2	0	0			
P2	9	0	2	3	0	2			
P3	2	2	2	2	1	1			
P4	4	3	3	0	0	2			

i) Calculate the need matrix.

ii) Is the system is in safe sate, if yes find the safe sequence.

Ans:

The need matrix for the above is

Need = max - allocation

Process	Need		
	A	B	C
P0	7	4	3
P1	1	2	2
P2	6	0	0
P3	0	1	1
P4	4	3	1

Step-2:- Calculate the safe sequence.

Make 3 columns. Assume available = 1

Finish	i	work
False	0	[7 4 3] >= [3 3 2] No, P0 wait----
False -> true	1	[1 2 2] <= [3 3 2] Yes, P1 will execute after execution it will release the resource by updating the availability available = available + allocation Available = [3 3 2] + [2 0 0] = [5 3 2]
False	2	[6 0 0] >=[5 3 2] No, P2 will wait.
False -> true	3	[0 1 1] <= [5 3 2] Yes, P3 will execute Available = [5 3 2] + [2 1 1] = [7 4 3]
False -> true	4	[4 3 1] <= [7 4 3] Yes, P4 will execute. Available = [7 4 3] + [0 0 2] = [7 4 5]

Those processes have been left in the 1st interaction, again hey will continue the same process.

P0 false ->true	0	[7 4 3] <= [7 4 5] Yes, P0 will execute available = [745]+[010]=[755]
P2 False-> true	2	[6 0 0] <= [7 5 5] Yes, P2 will execute Available = [7 5 5] + [3 0 2] = [10 5 7]
True	3	available = [10 5 7]
True	4	available = [10 5 7]

The order of execution of process =
<P1,P3,P4,P0,P2>
Safe sequence = <P1,P3,P4,P0,P2>

Q.

Process	max	allocation	available
P0	7 5 3	0 1 0	[3 3 2]
P1	3 2 2	2 0 0	
P2	9 0 2	3 0 2	
P3	2 2 2	2 1 1	
P4	4 3 3	0 0 2	

what would happen in P1 request one additional resource of instance a type and 2 additional resource of instance C-type.

Q:

Process	allocation	max	available
	ABCD		
P0	0012	[0012]	[1520]
P1	1000	[1750]	
P2	1354	[2356]	
P3	0632	[0652]	
P4	0014	[0656]	

Answers the following qu. Using bankers algorithm.

i) What is the content of need matrix?

ii) If a request from process P1 arrives for 0420 can request be granted immediately?

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.

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.

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 broke or not.

If YES, then it is ok i.e. deadlock is eliminated. If NO, abort the process from the deadlock statethen check.

Continue this process until we recover from deadlock.

This is also expensive method but better than first one.

- In this method there is an overhead because every time the DL detection algorithm is invokedafter 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 resource pre-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 from same 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 more than that, otherwise it create a starvation.

Unit-6

File Management

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

There can be a numeric data file, an alphabetic data file or alphanumeric and binary data 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 can store 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 the file 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 in a 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 the pointers (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 take 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 disk space.
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 no longer 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.

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 the file system.

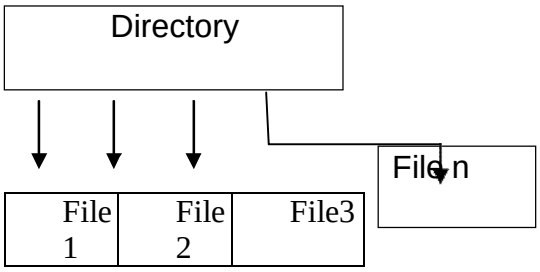
Different operations on the file directories:-

- 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 thedirectory.
- 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 can change.
- Traverse a file:- if we need to access every directory and every file with in thedirectory 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 all files
  - All files contained in the same directory name.



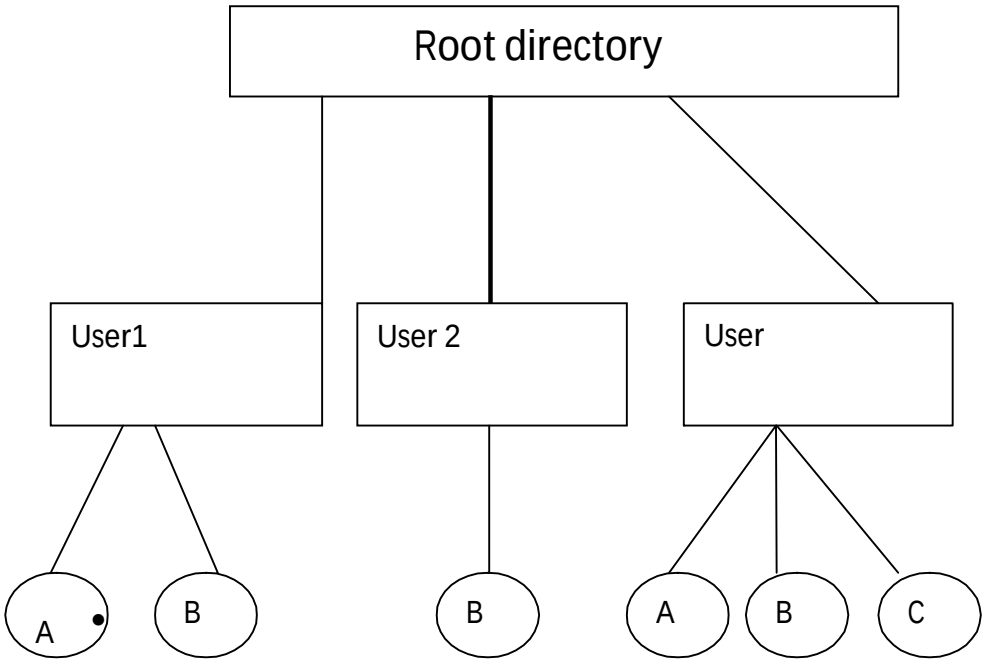
Advantage:-This scheme is very simple and ability to locate files easily.

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 use thesame name for their files.
- To avoid this problem each user need a private directory, so that name chosen by 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:-

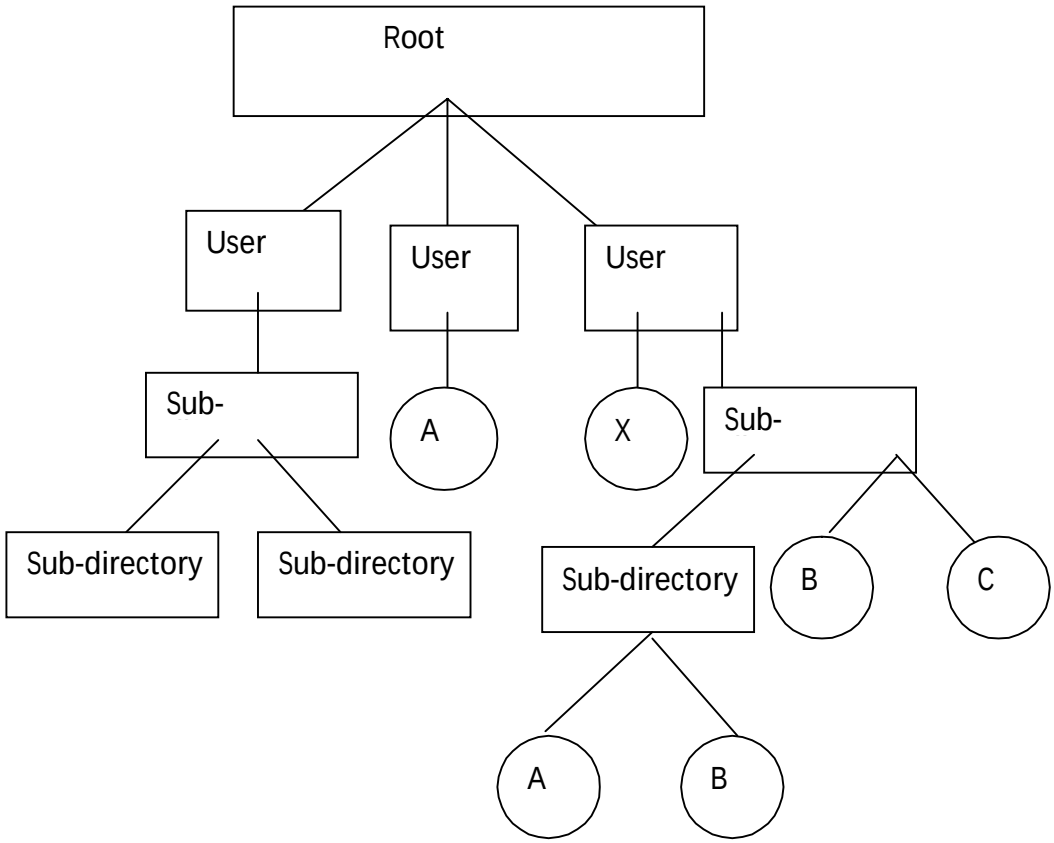


- Here root directory is first level directory. It consists of entries of "user directory".
- User level directories are user1, user2, user3 and it contains A, B, C files.

Tree structured directory/hierarchical directory system:-

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

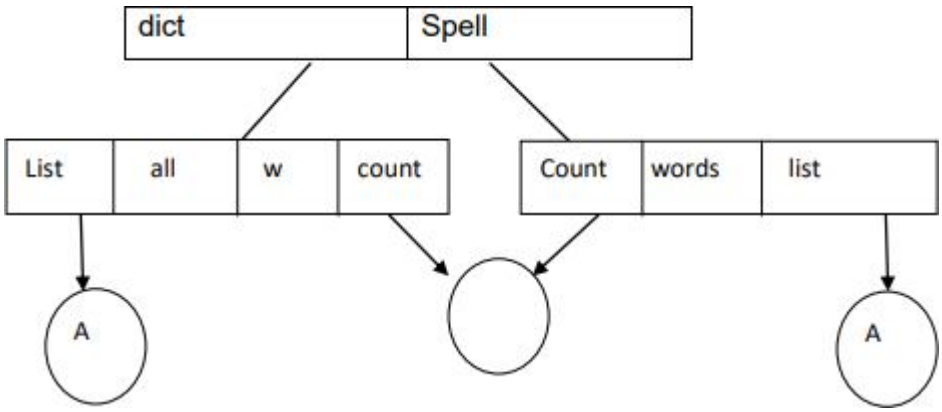
- Consider the following figure:-



[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 directory scheme but here cycle informed.



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 isto 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 andfunction

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

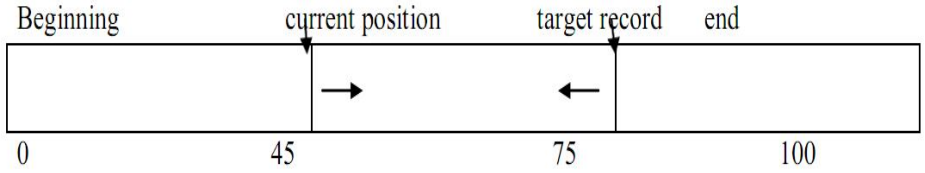
Files are used to store data. The information present in the file can be accessed by various access methods. Different system uses different access methods. Following are the most commonly used access methods:

- Sequential access

- Direct access
- Indexed sequential access.

Sequential access method:-

- This method is simplest among all methods. Information in the file is processed in 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 parallelprocessing.

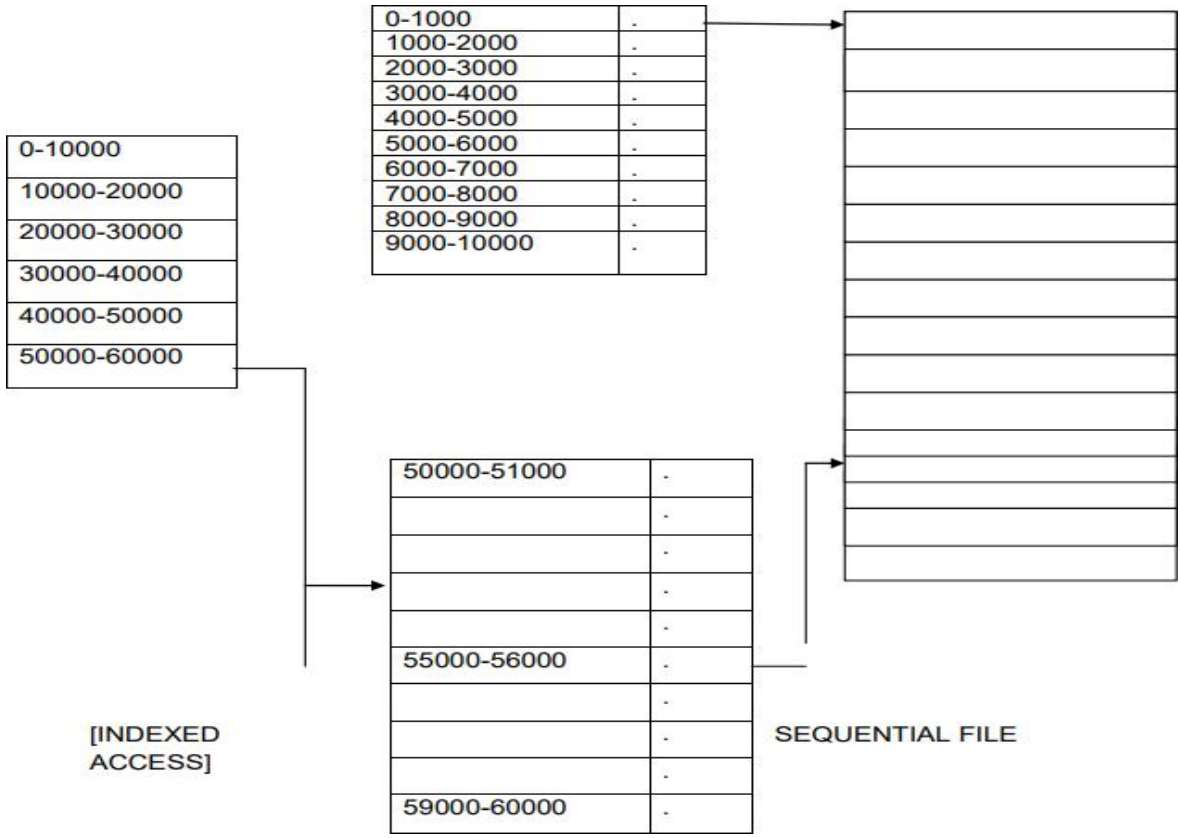
Direct access:-

- 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, disk allowsrandom 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 we want 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 a record. To overcome this problem, we can use this method.

- In this method (ISF), the records are stored sequentially for efficient processing. But, theycan be accessed directly using Index or key field. Keys are the pointer which 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 the total 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- Akey field
  2- A pointer field
- Suppose we want to access the 55,550th record, then the file management system (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 is neither purely sequential nor purely direct access.
- Generally indexed files are used in air line reservation system and payroll system.

File directories:-

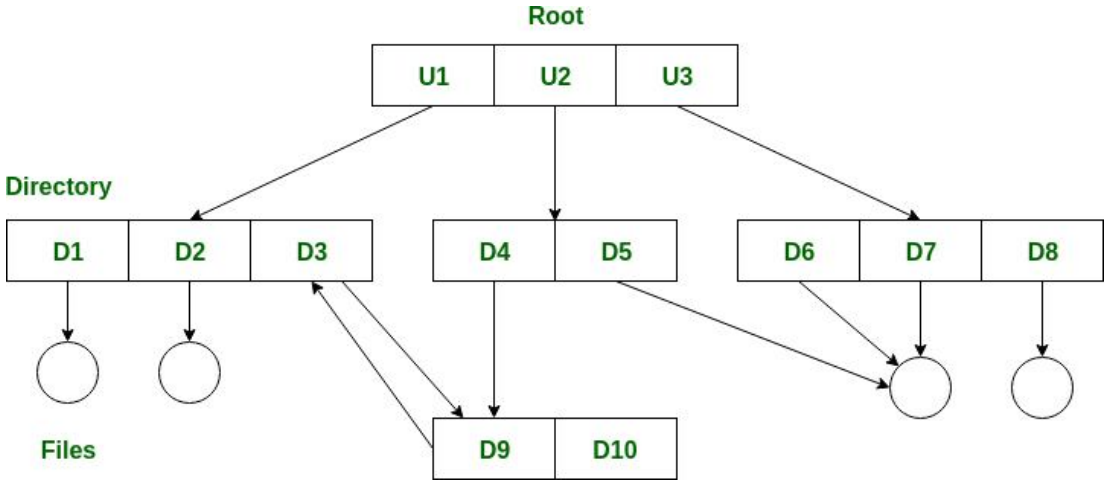
- The directory contains information about the files including attributes locations and ownership.
- Sometimes the directories consisting of sub-directories also.
- The directory is itself a file and it is owned by the operating system.
- It is accessible by various file management units.

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 isdestroyed, resulting a simple graph structure.
- The primary advantage of this structure is traversing is easy and file sharing alsopossible.



File allocation method:-

Files are normally stored in the disks so the main problem is how to allocate to these files 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 allocation

2- 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 100consecutive 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 block of thefile and size of the file.
- This method is best suited for sequential files.

Advantage

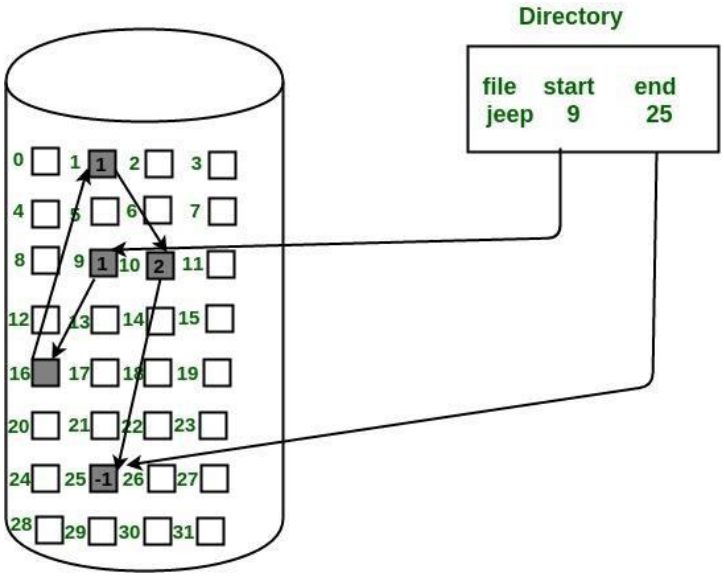
- Is avoid external fragmentation.

Disadvantage:

- It is difficult to find the contiguous free blocks in the disks.
- External fragmentation occurs (i.e:- some of the free blocks may left between 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:

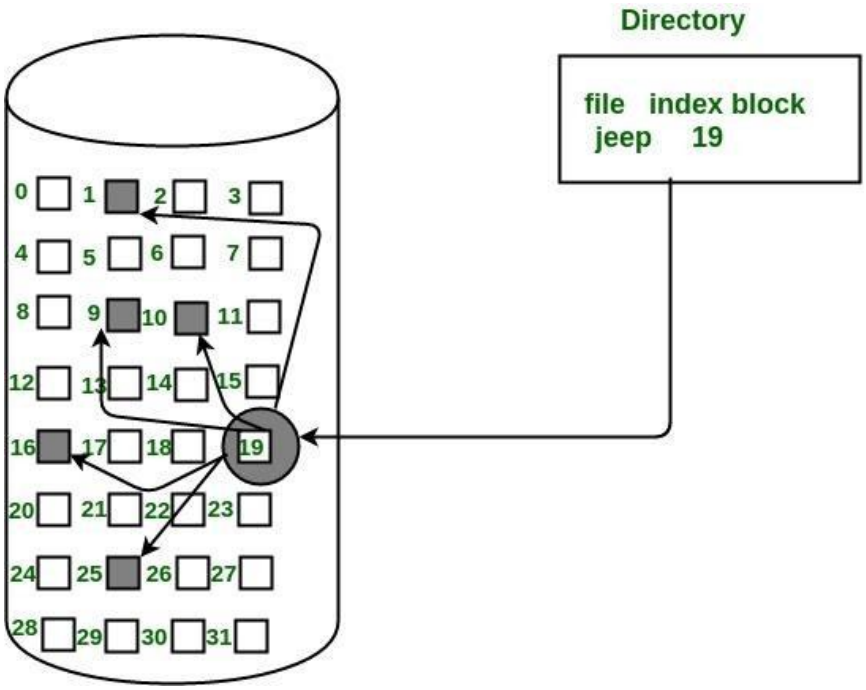
- Avoid external fragmentation.
- Suited for sequential files.

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 is known 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 lock 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 the index.
- No external fragmentation.

Free space management (or) disk space management:-

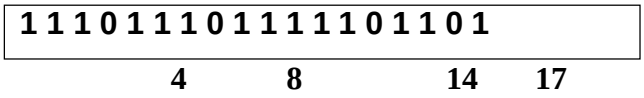
Generally the files are stored on disk so management of the disk space is a major problem to the designer, if user wants to allocate the space for the files we have to know what stocks on the disk available.

- 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 addedto the free space list.
- These are the number of techniques used for disk space management:- 1:-bit sector or bitvalue. 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 by one 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



Chain free points or linked free space list:-

- Another approach is to link all the free space blocks together keeping a pointer to the 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 the firstfree block.
- The n-1 of these are actually free.
- The last one is the disk address of another block containing the address of another '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 physical damage and improper access.

- Files can be damaged due to h/w problems such as temperature and validation and 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 floppy disks and storing them at a safe place.
  -but in multi-user system, there are various mechanism used to provide protection.

They are:-

1- Type of access
2- 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 can accept orreject 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 their accesscontrol.
- When a user requests an access for file, then first it checks the access list relatedto 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 it is very difficult to maintain and construct a list.
- In order to solve this problem, access control is introduced. The system classifies the 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 to access.
- 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 makes all thefiles accessible.

UNIT-7

System programming

System programming:- System programming is the activity of programming system software.
The system programming aims to produce software and software platforms which provide services to other software.
System programming requires a greater degree of hardware awareness.
Whereas the application programming aims to produce software which provides services to the user directly.

Application program:-

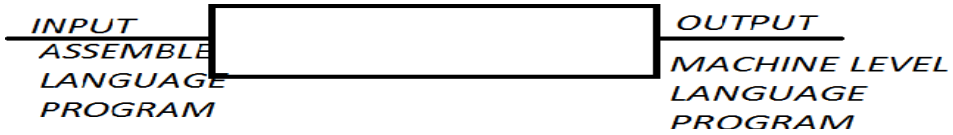
1. Application software is a set of one or more than one programs which are designed to carryout operation for a specified application.
2. For example payroll packages are designed to produce pay slip as the major product. An application package for processing examination result produced mark sheet as the major product.
3. 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 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:

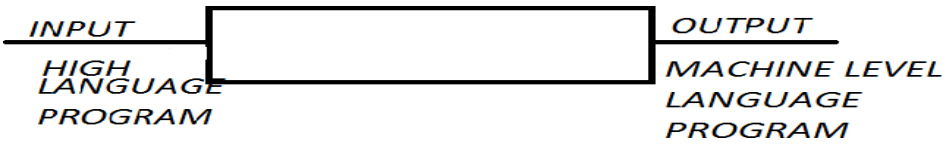
1. A computer program which translates an assembly language program to its machine language equivalent is known as assembler.
2. The assembler is a system program which is supplied by the manufacture.
3. A symbolic program written by a programmer is called a source program. After the source program has 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 output from the assembler is the machine language program.

Compiler:

1. A complier is a program that translates the high level language into machine 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 a complier is referred to as object program.



3. So input to a complier is known as source program and output from a complier is known as object program.
4. A single complier cannot translate all the high level language into machine level language should have a dedicated complier for its compilation.
5. Complier is a large program which resides in secondary memory. When it is required it is copied into main memory.

Interpreter

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

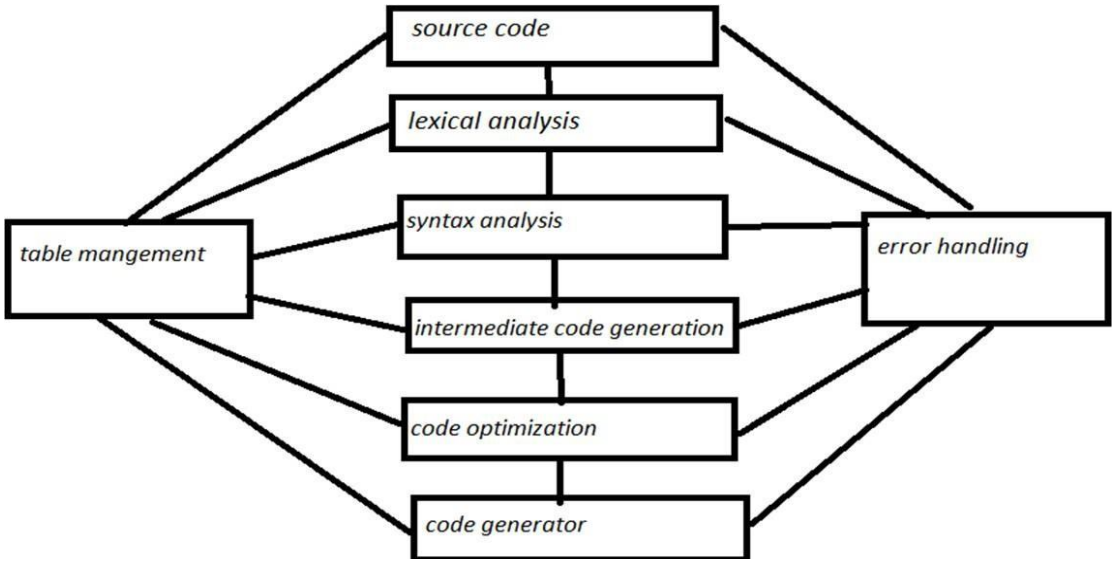
Here translation and execution alternate for each statement encountered in high level language program.

An interpreter translates the instruction and the control unit executes the resulting 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 that translates 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
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 toform 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.