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)

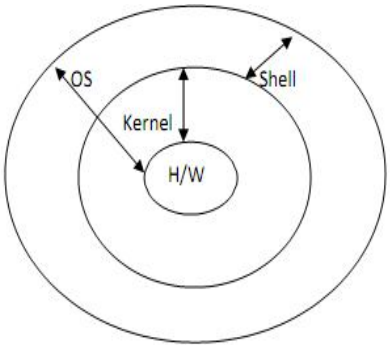
Protection/Security Management:

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 divides 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.

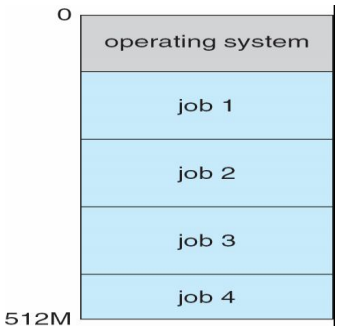
Advantage:

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

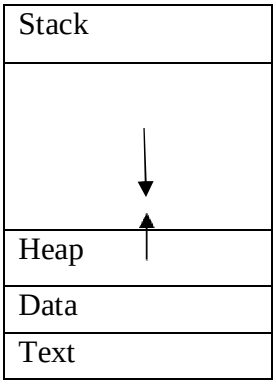
PROCESS:

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.

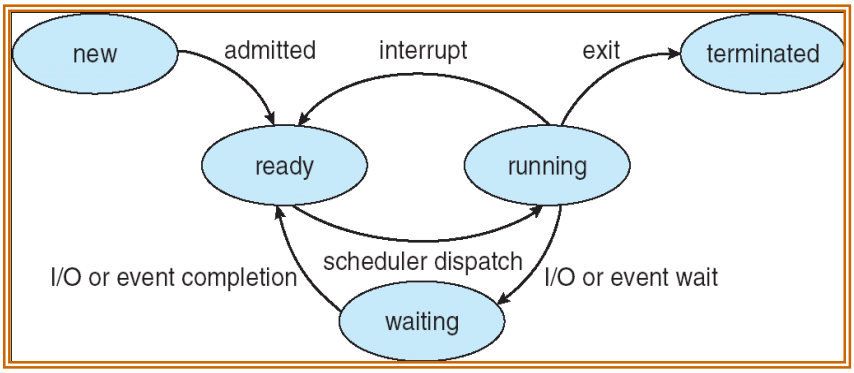


Diagram of process state

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.

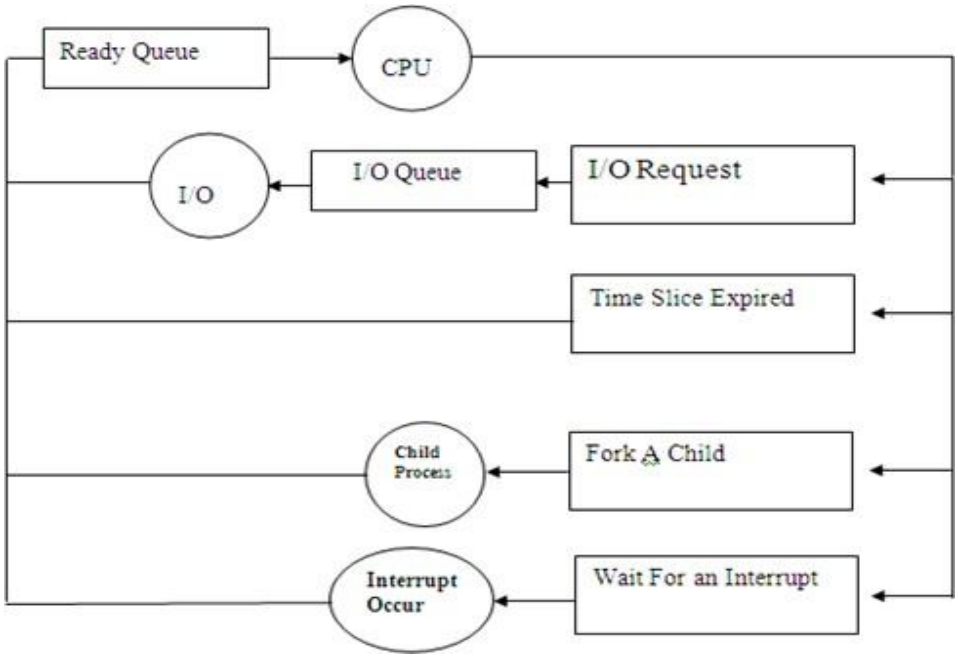
Device Queue:

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.

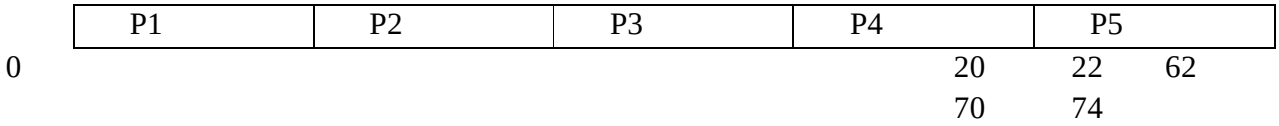
Let the process arrives in the order p1, p2, P3, p4,p5.

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 tocommunicate 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 thefunctioning of other processes.

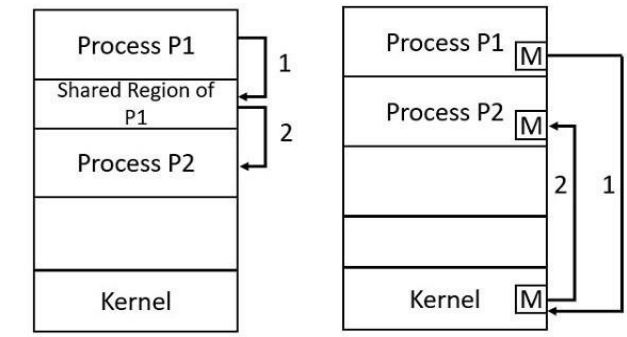
Co-operating process:-

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 allowprocesses 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?