23/02/202
1st hour

Operating System

Computer = Software + Hardware
(CPU, RAM, ROM, I/O devices)

↓

Set of tested programs + documents which are used to work. an application

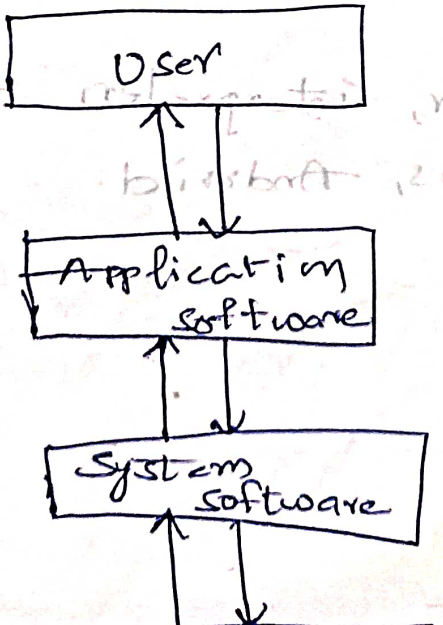
A computer is a combination of both hardware and software

Hardware :- The electronic devices RAM, ROM, I/O devices are assembled with a help of cables is called hardware.

Software :- The set of well tested programs and the document that says how to maintain and how to operate is called software.

There are two types of softwares as

1. Application software
2. System software.



Application software :-

The set of programs written for satisfying the specific task is called application software

Eg :- Editplus, Notepad, media player, ~~Compiler~~, ~~interpreter~~ ... etc.,

To install the application software computer must contain an operating system

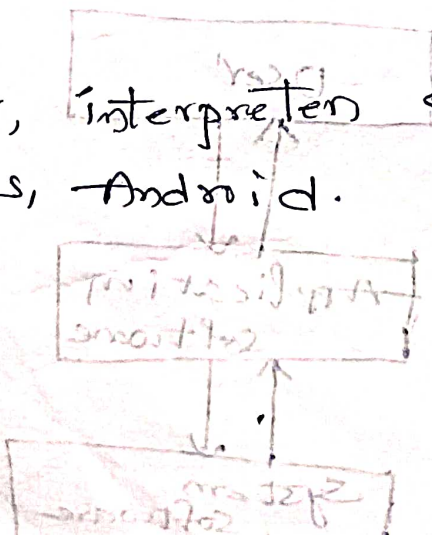
System software :-

The programs that are written for operating the hardware components is called as System software.

These programs written for general purpose that should be efficient and consistence to all the users

Without system software we can't operate Computer

Eg :- Compiler, interpreter debugger ... etc.
windows, Android.



Functions of operating systems

1. Resource Manager
2. process Manager
3. Memory Manager (Main)
4. I/O Manager
5. Storage Manager
6. Security and protection

GUI is an interface which performs system calls and internally calls well defined methods

Types of operating system

1. Batch operating system / Non-multi

with respect to generations of the computer there are so many changes in designing the OS. when we see the OS from 1st generation to today. The types can be

1. Batch OS
2. Multi programming OS
3. Multi tasking OS
4. Multi processing OS
5. Real timed OS.

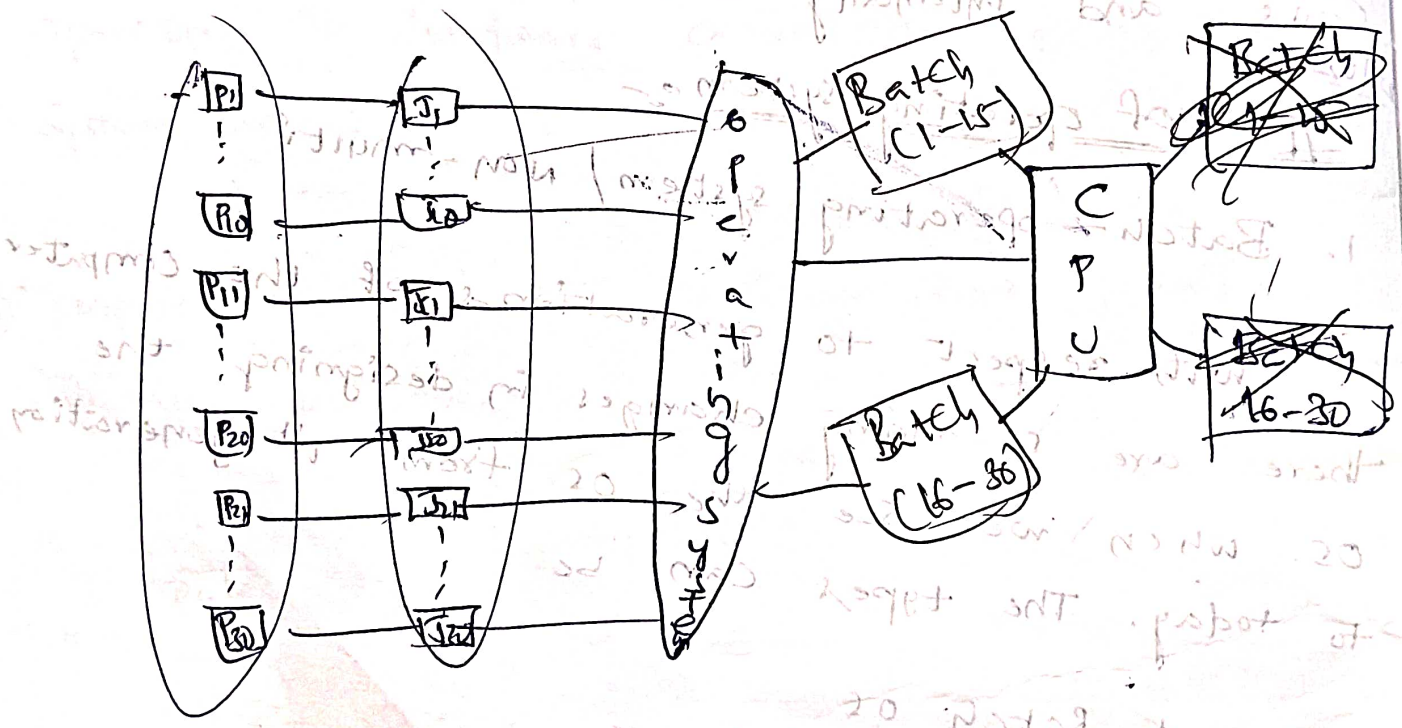
1. Batch OS :—

It is an type of OS that allows multiple users to use it at same time, without direct communication between them.

This is done by having the users submit their jobs to the operating system, which then processes them one at a time

Ex :- payroll, Bank statements

let us consider 20 users doing job on OS then it works on CPU at a time of allocated



advantage :—

Easy to estimate the time to Complete a given job, easy to manage large repeated work, idle time is less sharing by multiple users is possible

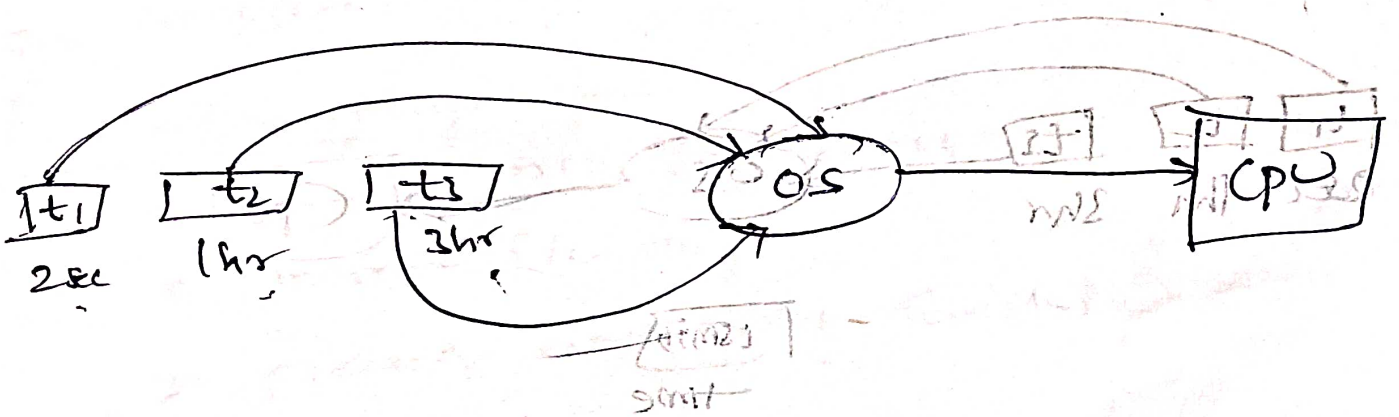
disadvantage :— 1. if a job fails the other

Costly

2. Multiprogramming OS :-

It is an type of os that allows multiple task are going to happen on a os with direct interaction/ communicat

here, let us consider three task task 1 takes 2sec of time, task 2 takes 1hr, task 3 takes 3hrs. Task 2 ; execute first then we have to wait 1hr to execute another but here time space of task one is 2sec. but these has to wait.



advantage :-

Direct interaction between task and os

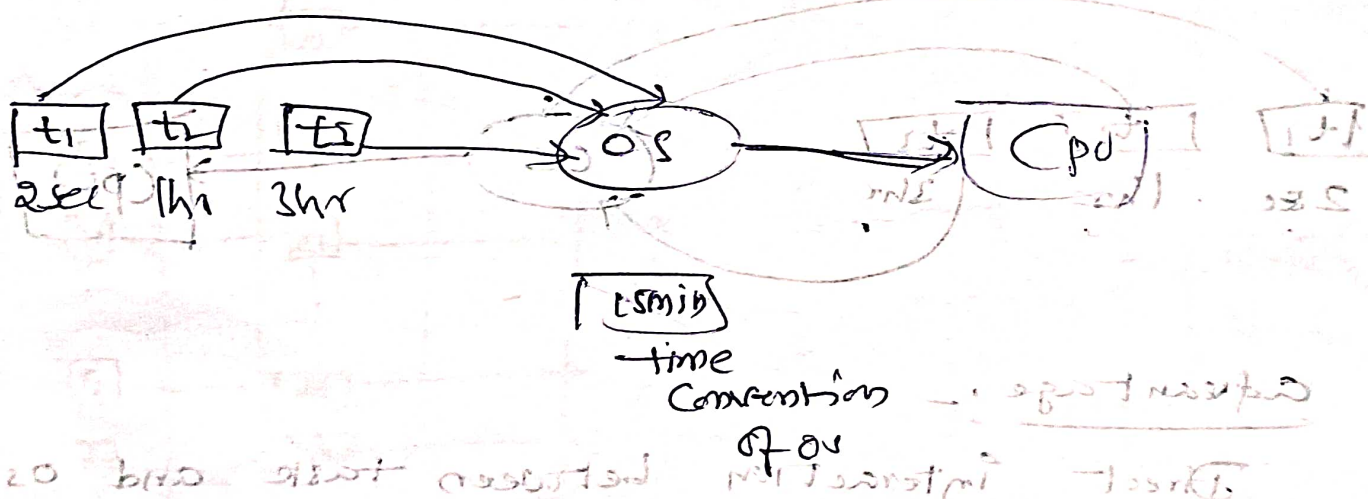
disadvantage :-

* it takes time complexity.
* waiting time.

of Multi programming OS; The multitasking OS is discovered.

Here also there is an direct interaction between task and OS. there is time convention for each task there is time convention for certain time. After time excluded then next task goes until then previous task waits in Queue.

Ex:-



* if t_2 goes first to execute

$t_2 = 1hr - 15min(OS) := 45min$

Queue:- [| | | t_3]

* if t_1 executes second,

$t_1 = 2sec - 15min = -14.58sec.$

These 14.58 sec can't waited

these time is used to complete next task to reduce waiting list

* t_3 executes then

$t_3 = 3hr - 15min = 2hr\ 45min$

Queue :- | | | t_2 | t_3 |

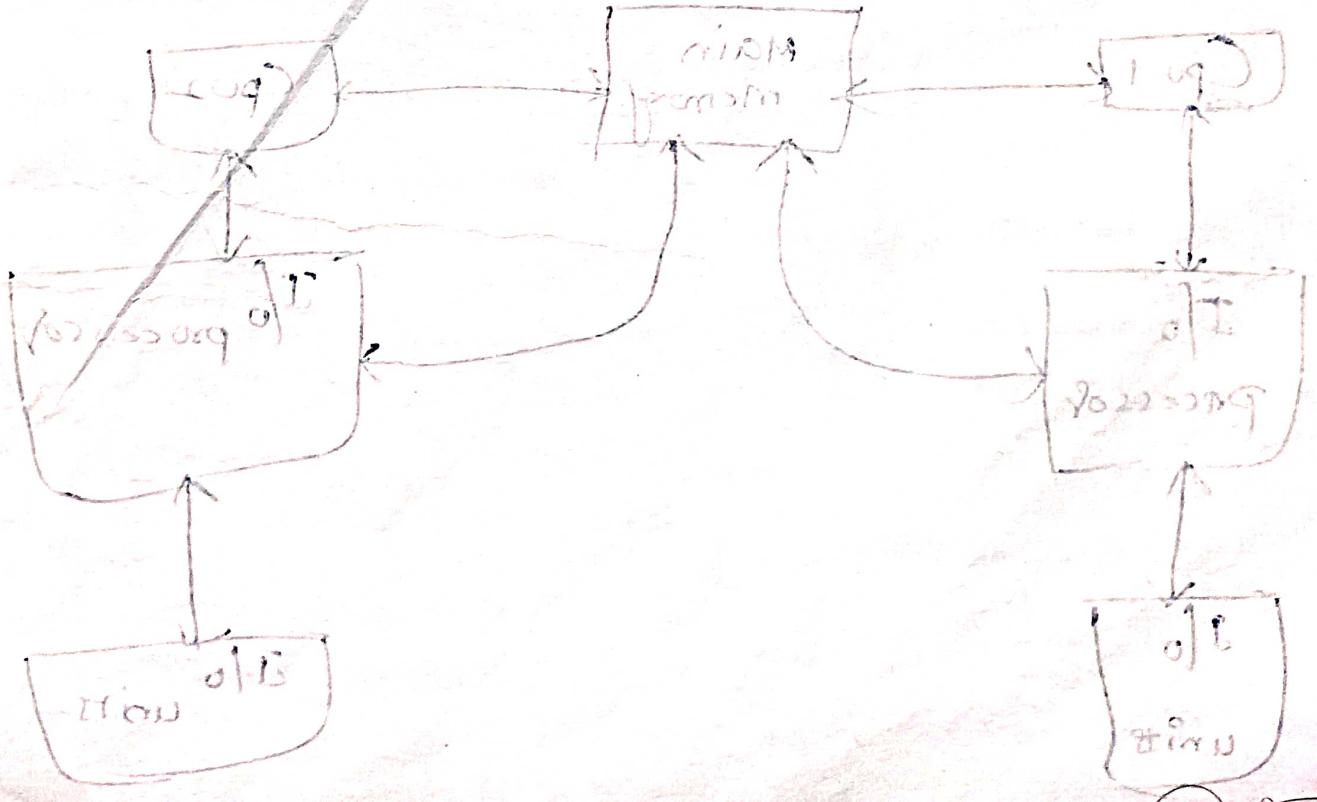
These will repeated until the program task completed.

Advantage :-

* waiting time reduced
* time convention is big advantage to complete small time programs

disadvantage :-

* Greater risk of errors
* lower efficiency
* processor slows very Quickly

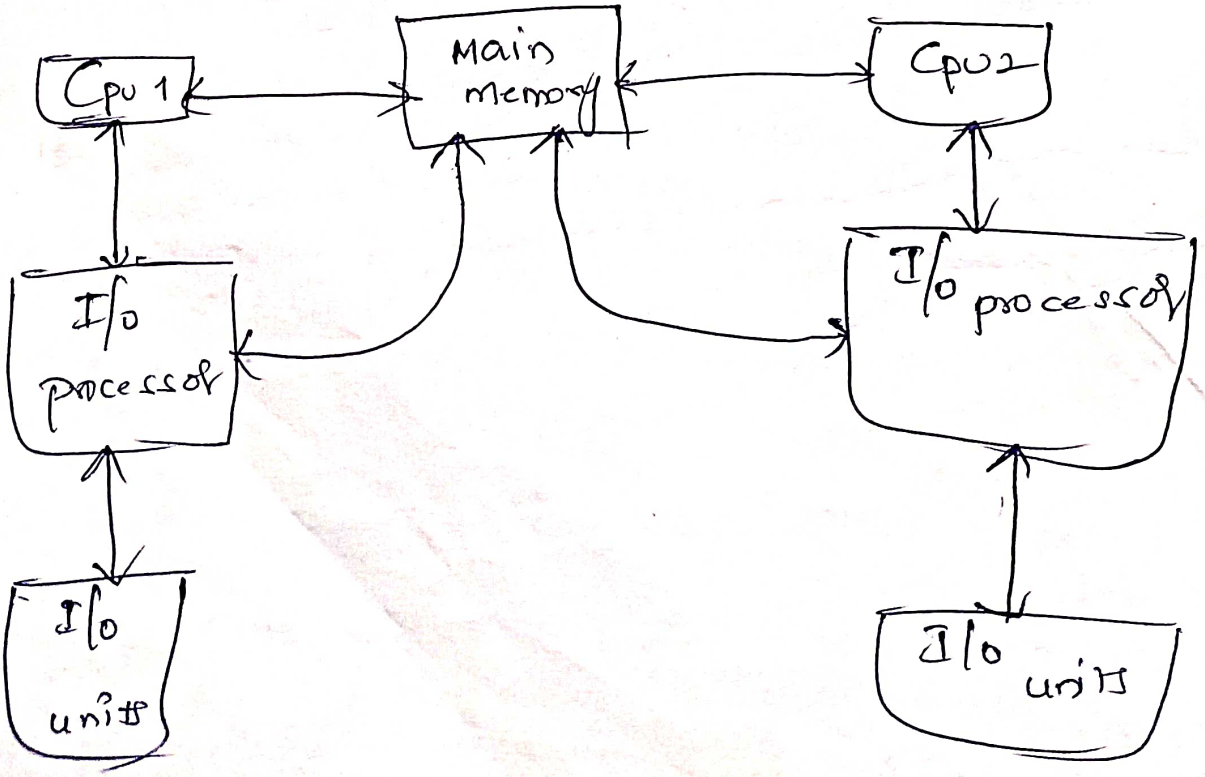


Multi-processing OS :-

Multiprocessor OS are used in OS to boost the performance of multiple CPUs within a single Computer system.

Multiple CPUs are linked together so that a job can be divided and executed more Quickly.

* Parallel computing is performed by multi processing
* There are several processors in a system and each of them can run multiple process Simultaneously



Types of multiprocessing OS :—

1. Symmetrical :— It is a computer processing done by multiple processors that share a common OS and memory

2. Asymetrical :— An asymetrical multiprocessing system is a multiprocessor computer system where not all of the multiple interconnected central processing units are treated equally.

Advantages :—

* Helps in parallel computing
* Reliable
* Increased throughput
* Economical

Disadvantage :—

* More Complex architecture
* Required large memory
* The system crashes if the processor in charge of a particular task fails.

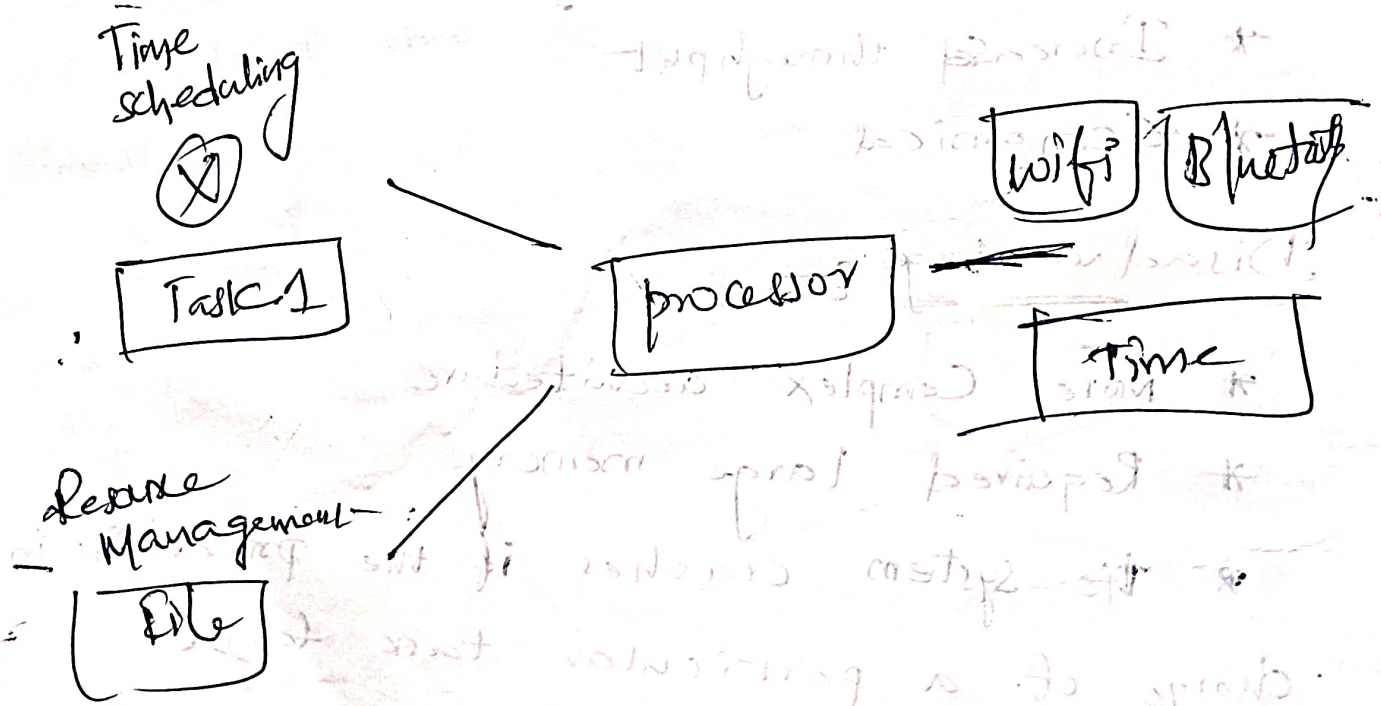
Real time OS :-

It is an OS that guarantees real-time in computers that has performed. It is employed mostly in those systems in which the results of the computations are used to influence a process while it is executing

Advantage :-

→ Easy to layout develop and execute real-time applications under the real-time OS.

→ memory allocation is best managed.



Operating systems operations :-

ROM — Post
Bios
Setup

In every operating system, there are previlised programs and secured programs. This acts as a layer which is called Kernel. The Kernel program is directly interacting with the hardware Component.

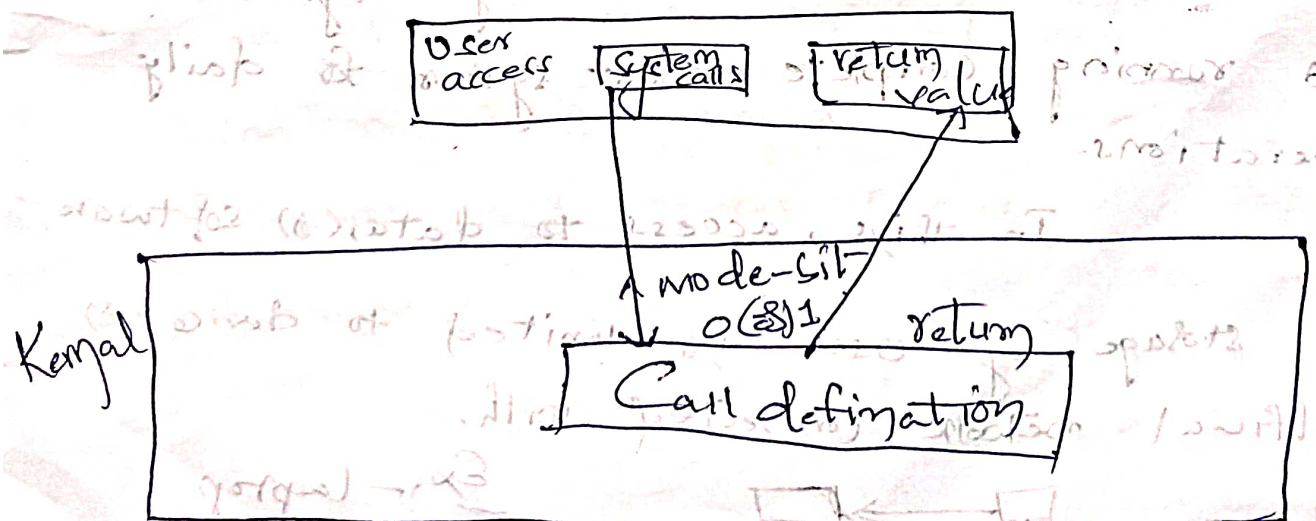
In Kernal there is a mode-bit which is a hardware component having value either 0 (or) 1.

Dual-mode :-

The user mode and Kernal mode together is called dual-mode. either user (or) Kernal mode communicates with the CPU to complete the task.

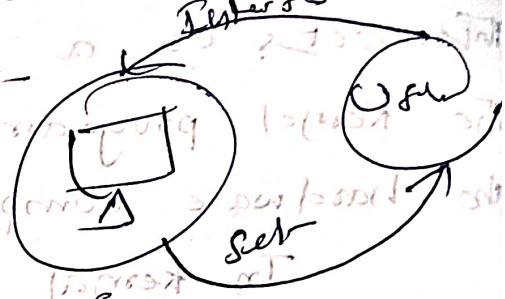
When user mode is going on, the mode-bit value is 0 and 1 for Kernal mode.

Sometimes without Kernal OS can interact with the CPU



Timer mode :-

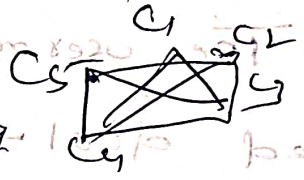
Another operation of the OS is timer which starts the counter value and reaches to the set value with interput the user with the error msg (or) alert msg.



Computing environments :-

There are 8 computing

1. Traditional / wired
2. Mobile / wireless
3. distributed / parallel - stuff - or - clusters
4. Client - server
5. Peer - to - peer computing
6. virtualization / images
7. Cloud / platform, infra, fx
8. Real time embedded



1. Traditional :-

It is a process of using physical data centres for storing digital assets and running complete network system for daily operations.

In this, access to data, (&) software, (&) storage by user is limited to device (&) official network connected with.

2. Mobile :—

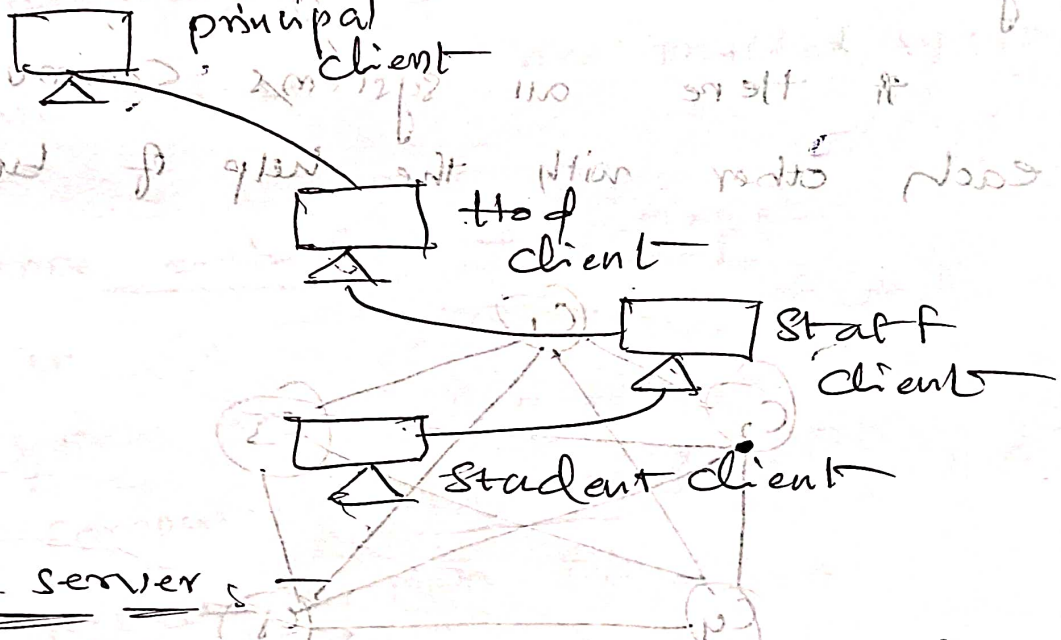
* It is an wireless communication
* It refers to various devices allow people to access data and information from where they are, when they are constantly moving (or) standing

Ex :— Cellphones

3. distributed :—

In this environment, all the systems/ clients are distributed the specific task to make the task easier and reducing time.

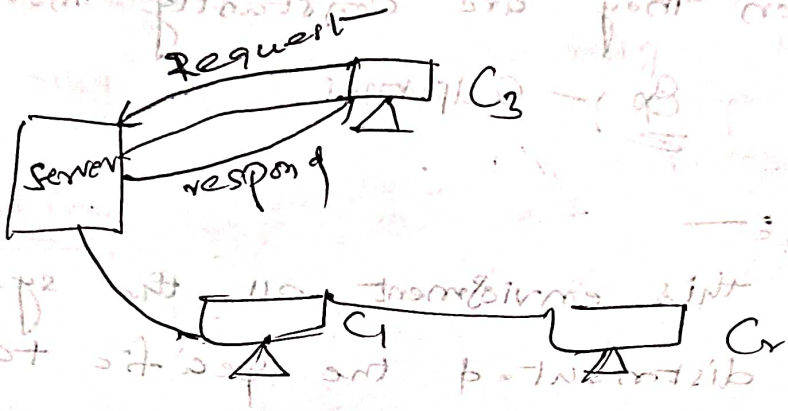
Ex :— university distribution



4. client — server :—

there all the clients are connected to a server with topology. if any client requires an information/ data then it request to server and server checks the request and respond.

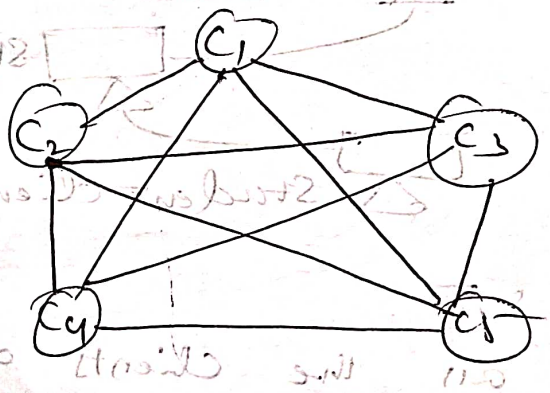
Ex :- There are 3 computer systems are connected to server. C_3 requires file from C_1 then C_3 request to server and server takes the information and respond to C_3



Peer-to-peer :-

* It is an direct communication b/w systems to retrive & give data b/w them

* Here all systems communicate with each other with the help of bus.



6. Virtualization :-

virtualization is the creation of virtual version of something such as server, a desktop, a storage device, an OS & network resource such maintain called as virtualization.

7. Cloud :-

Cloud computing consists of

* paas
* Iaas
* Saas

Investment

In this cloud computing the user is only have to invest and relax and all the required platform, Infrastructure and software are provided by cloud.

8. Real time embeded computing :-

It is a particular version of an embeded system that works on the basis of real time computing represented by a dedicate type of operating system RTOS.

If working principle are as follows: Quick response to external facts: an embeded system must work with in fixed time constraints.

2. Operating System structure :-

→ At the time of designing the operating system, the developers may have a clear idea on the services that should provide to the users/ programs

→ The operating systems categorized in 2 :-

1. System services
2. User services

User services :-

Whenever user interacting with OS it facilit the following services:

1) User Interface
2) program execution
3) Communication
4) I/O controller
5) Error detection
6) File creation & manipulation

System services :-

1) Resource allocation
2) Acounting
3) Security & protection

1. User Interface :-

* User and OS are connected with each other help of Interface.

* There are diff. Types of Interface.

Connection.

a. Command line execution :-

* When user needs different commands regarding to i/p and output & to perform tasks

* used to execute output and create, delete, print, copy, ...

* Basic operation performed in CLI

* Responsible for memory management

b. Graphical User Interface :-

* playing Games, videos done with GUI. These all need graphics.

* Taskbar showing running programs

Ex :- Linux and windows

c. program execution :-

* program always execute by processor with help of OS

* C drive is secondary storage device.

* program statements convert to binary code - processor.

* Refers the process by which the program is loaded into memory executed and then terminated.

* OS responsible for execution of programs

* Loading, Linking, Allocation, execution

Overall it is complex process that involves many step

munication :—

→ process communicates on another sub proc is called communication.

→ OS refers to exchange data between different processes with system.

4) I/O Controller :—

→ OS acts as device controller whatever Input output f/s attached to OS is I/O controller.

→ I/O f/s a hardware component that interfa various input & output devices like keyboard, mouse, printers, networks - - -

→ OS communicate with I/O through device drivers acts as interface b/w OS and I/O Controller.

5) Error detection :—

→ process of identifying & correcting errors (or) mistakes in a system.

→ This techniques are used to ensure the accuracy & integrity of data as it is transmitted (or) stored.

6) File Management :—

→ OS search the data to store.

→ Computer → process of → process, store & retrive the data

→ Refers to process

→ Security measures to protect files

System services :—

1. Resource Allocation :—

A Resource has many processes when P, allocate resource, then P2 is waiting system calchate by os operator.

→ The process of assigning available resources such as cpu time, memory. Ilo devices & network to completing process on system.

→ All process get a fair shape of resources

1. Round Robin
2. priority
3. FCFS
4. Short job first

2. Accounting :—

→ In multiuser it maintain all data that how many user are there, when they login & logout & what they do.

→ It enables system administrators to make informed decisions about resource allocation, system optimization and system security.

→ Tracks the usage of system resource by processor & user.

→ Security measures to protect

System services :—

1. Resource Allocation :—

A Resource has many processors when P, allocate resource, then P2 is waiting system calculate by os operator.

→ The process of assigning available resources such as cpu time, memory. I/o devices & network to completing process on system.

→ All process get a fair shape of resources

1. Round - Robin
2. priority
3. FCFS
4. Short job first

2. Accounting :—

→ In multiuser it maintain all data that how many user are there, when they login & logout & what they do.

→ It enables system administrators to make informed decisions about resource allocation, system optimization and system security.

→ Tracks the usage of system resource by processor & user.

3. Security & protection :-

→ unauthorized acess system. protect by passwd

It have many folders which has security

⇒ It store data & retrive data. performs Read/ write/ append for protection of file.

→ They require a combination of technical meas & user awaness to ensure the safety & security of system resources.

System Calls :-

System Calls are the interfaces in between operating system and its services.

→ The program under execution is called process at every process system Calls are executed.

→ The system calls execution starts at, chang the mode bit from 0 to 1 and involving the kernel code.

After completion bit value changes from 1 to 0

This is called Context switch.

→ The following are the four cases where the Operating system make system Calls.

1. Creating closing, deleting the files
2. At every process creation
3. At every new hardware component is attached

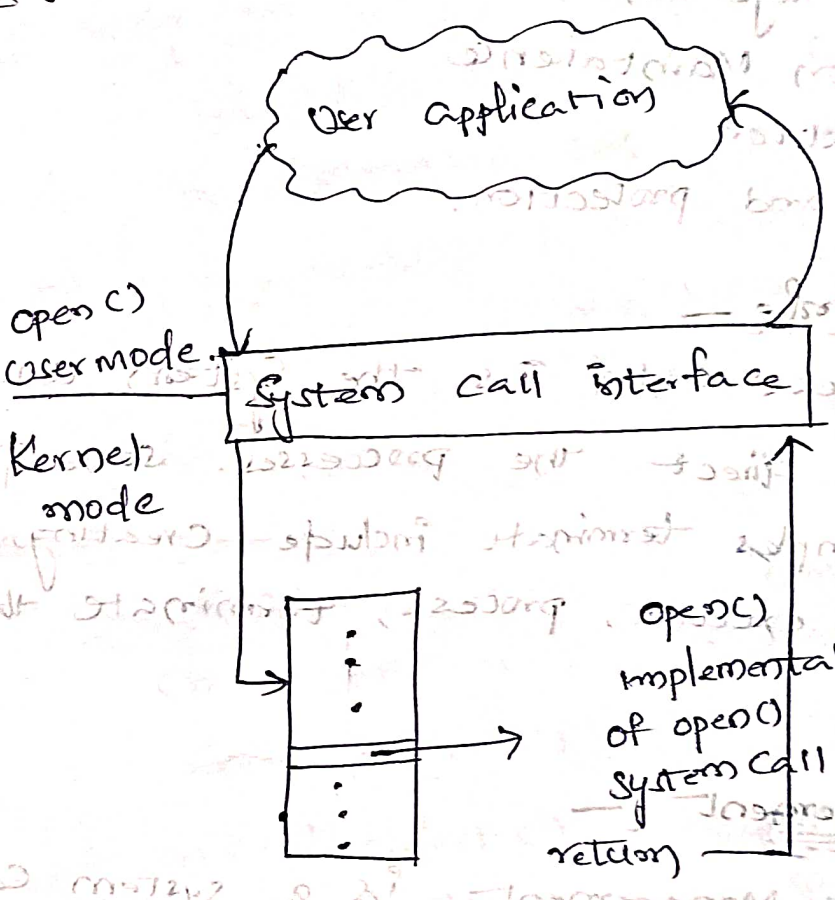
→ OS has its own API is to execute any Sourced language programs.

→ Win 32 API for windows application.

→ JVM API for Java programs.

→ POSIXDS API for Machintosm.

API System Calls – OS Relationship :-



→ System calls have a unique identify that has seen indexed in the table

→ Based on the priority of index Kernel calls System methods

→ The priority can be 13, 12, 11, - - - -

→ For every system method there is a well-defined code for producing the response to the user process.

→ Sometimes the system calls can there parameters interms of registers, tasels (or) stack.

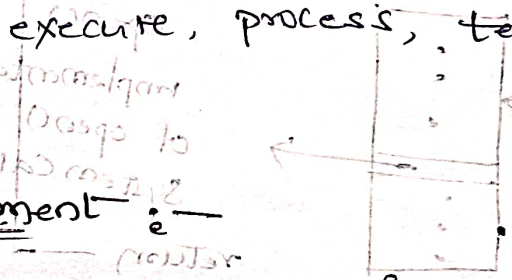
→ parameters from tasle (or) stack has no limit.

Types of system Calls :—

1. process Control.
2. File Management
3. Device Management
4. information Maintaience
5. Communication
6. Security and protection.

1. process Control :—

process control is the system call that is used to direct the processes. some process Control examples terminate include creating, load, about, end, execute, process, terminate that processes.



File Management :—

File Management is a system call that is used to handle the files. Some file Manage examples include creating files, delete files, open, close, read, write etc.,

Device Management :—

Device Management is a system call that is used to deal with devices some examples of device Management include read, device, write, get device attributes, release device etc.,

Information Maintaince :-

Information Maintainence is a system call that is used to Maintain information. There are some examples of information maintaince including system data, set time & date, get time & date, set system data etc.,

Communication :-

Communication is a system call that is used for Communication. There are some examples of communication including create, delete, communication connections, send recieve message etc.,

Processes :-

Windows	Linux
Create process()	Fork()
Exit process()	Exit()
Wait for single object()	wait()
Create file()	Read()
Read file()	Write()
	open()
Close Handle	Close()

Security and protection :-

protection and security requires that computer resources such as CPU, softwares, memory etc. are protected this extends to the OS as well as the data in system.

System Programs :-

* It provides environment to process the user request (OS) services by executing well defined code

* We know there are two types of softwa

1. Application :- program directly interact with user

2. System :- program that gives servic to another software.

* The system ~~software~~ programs are :-

1. Status information
2. Communication
3. File manupulation
4. File modification
5. program loading & executing
6. programming language suppo-t
7. Application programs.

1. Status Information :-

It gives the status information of an drive. That means it gives an overall idea about the task bar.

Eg :- To check the space availability in a drive

let us open drive bar and select the drive status you want and go to properties. of the drive we get the total status information.

2. **Communication** :-

communication is used to provide access to systems and exchange files and message in text, audio and viedo format between Computers / users.

The communication is eg mail / e-interest etc., This communication consists of set of instructions and information.

user to user
user to prosser
prsr to user

3- **File manupulation** :-

Firstly, we required a file to create/ already created file. This file we want to perform specific operations on them called as file manupulation.

The operations we are going to perform

1. **Create()** :- Create is used to create a file.

2. **Copy()** :- Copy is used to duplicate the same file with same feautres and properties in another location / same location.

3. **Moving()** :- Moving is nothing but cut and paste that means same file is moved from one place to another.

4. File Modification :-

The created file is going to perform specific operation inside / internally called as file modification.

The specific operations are :-

1. add :- adding the required data as an internally
2. open :- To open the file
3. remove :- to remove unwanted data.

5. program loading & executing :-

* placing a program into main storage is called loading the program.

* To load and execute a program, use either the LOADGO (or) CALL command.

* LOADGO command loads objects and load modules produced by a compiler (or) assembler and program objects and load modules produced by the binder (or) linkage editor.

6. program language support :-

* systems programmers design and write systems software.

* For ex, they might develop a Computer's OS, such as

* Although Java and python are the great languages for system programming
* Each language we have to install externally and we have to fix path.

7. Application programs :—

A type of programming that is to develop applications that interact with system software. (OS) or computer hardware.

Ex:- Ms-word, excel, powerpoint etc.,

OS - debugging :—

Debugging is the process of finding the problems in a computer system and solving them.

* This is cost-effective because it has to be solved after deployment.
* The operating system performed debugging by using log files

log files :—

These record all the events occured in OS. This is done by writing messages in a file.

There are different types of log files

1. Event logs
2. Transaction logs

3. Message logs

Event logs :-

These stores all the records of the events at the time of execution of a system.

Transaction logs :-

These logs stores changes of the data so that system can recover from crash and other errors.

Message logs :-

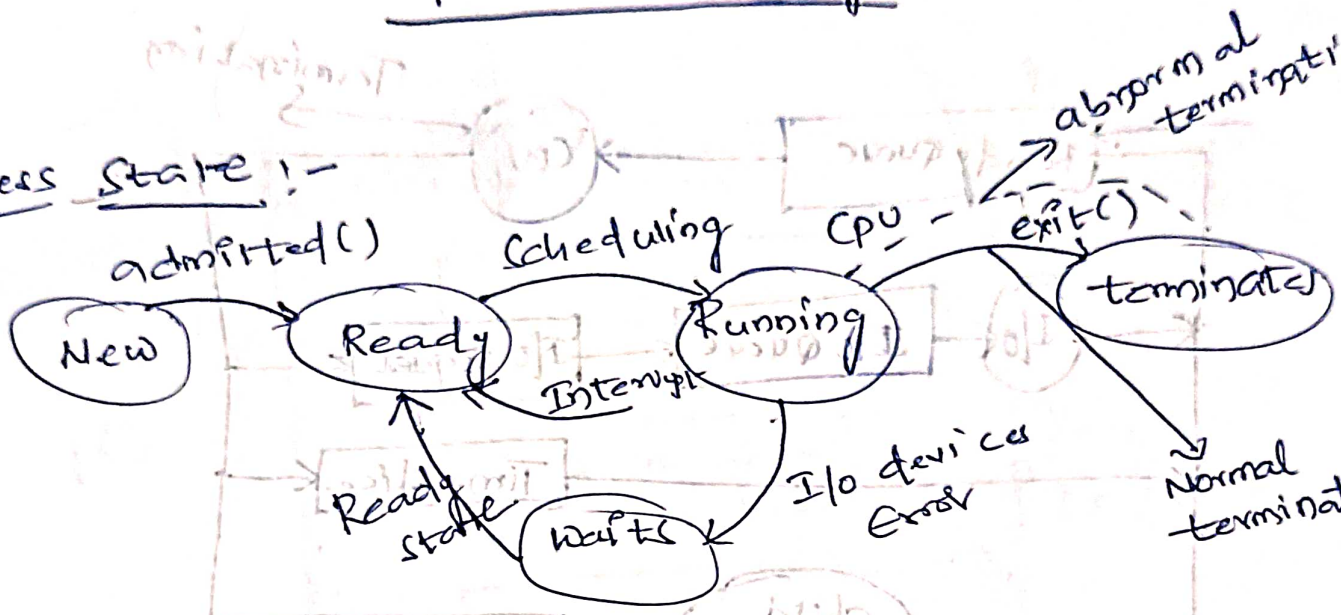
These are public and private, they are mostly plain text files and some are html files.

10/03/2025

Unit - II

Process Scheduling

Process State :-

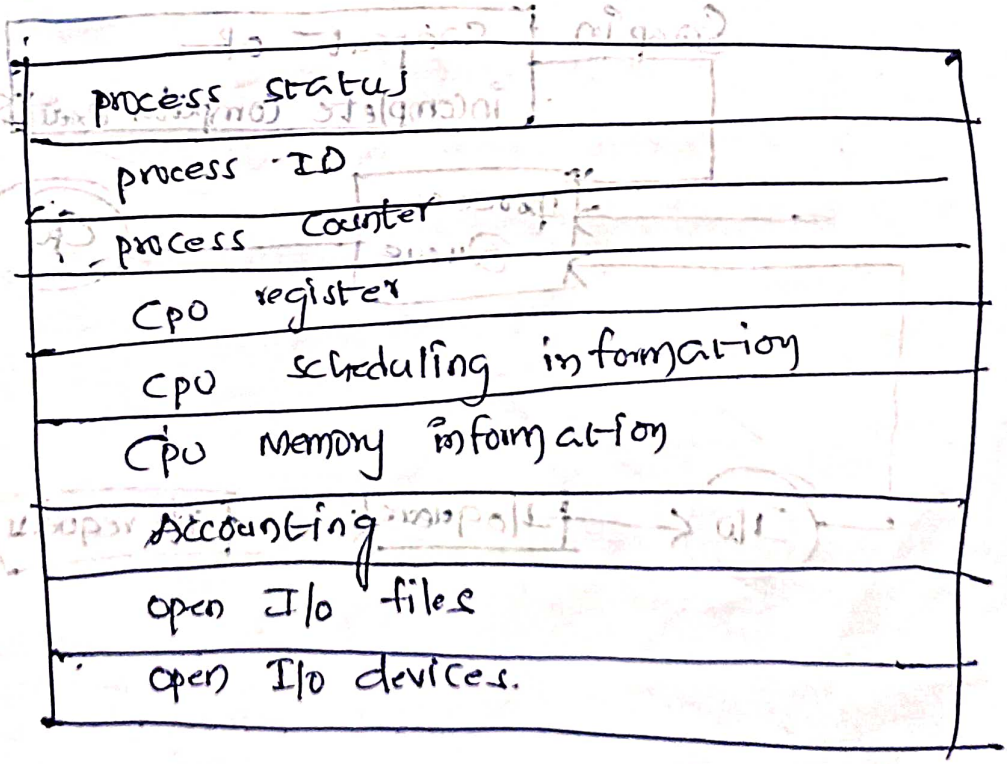


The circle denotes states.

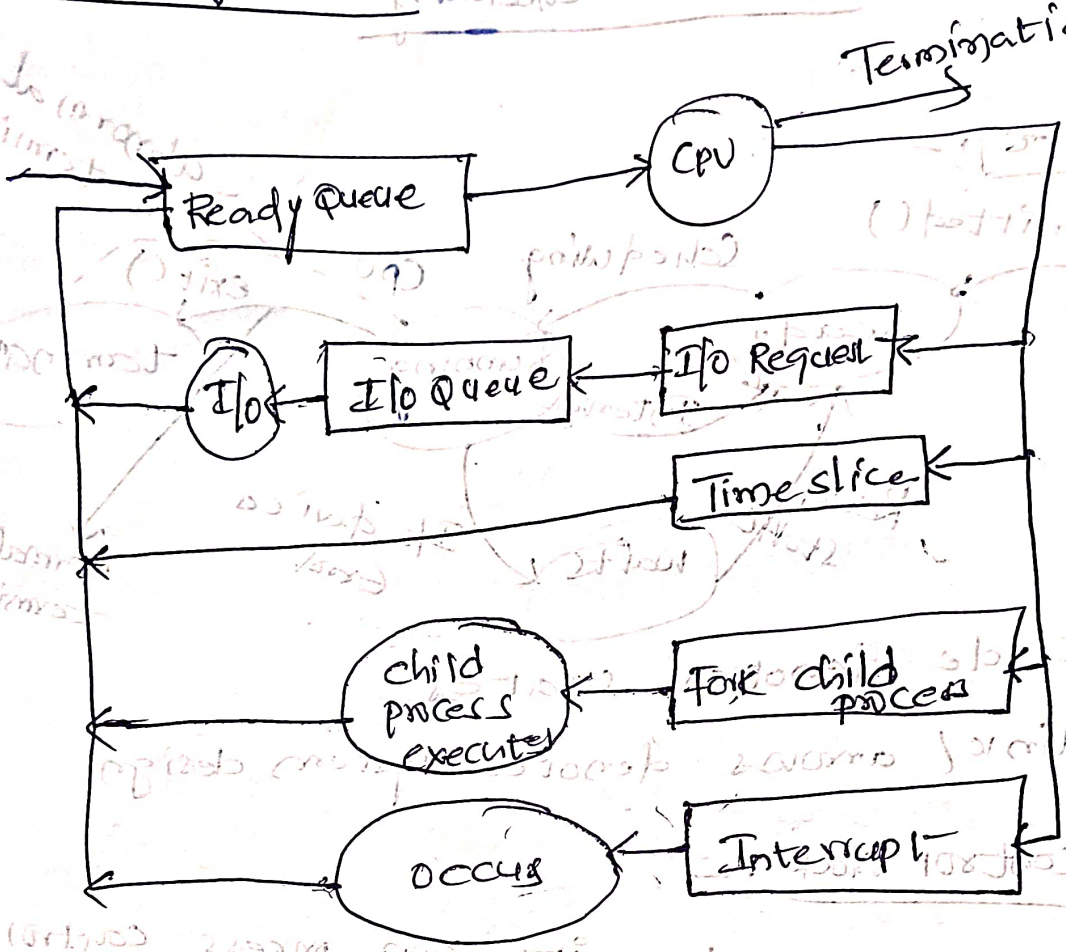
The link / arrows denotes system design.

Process Control Block :-

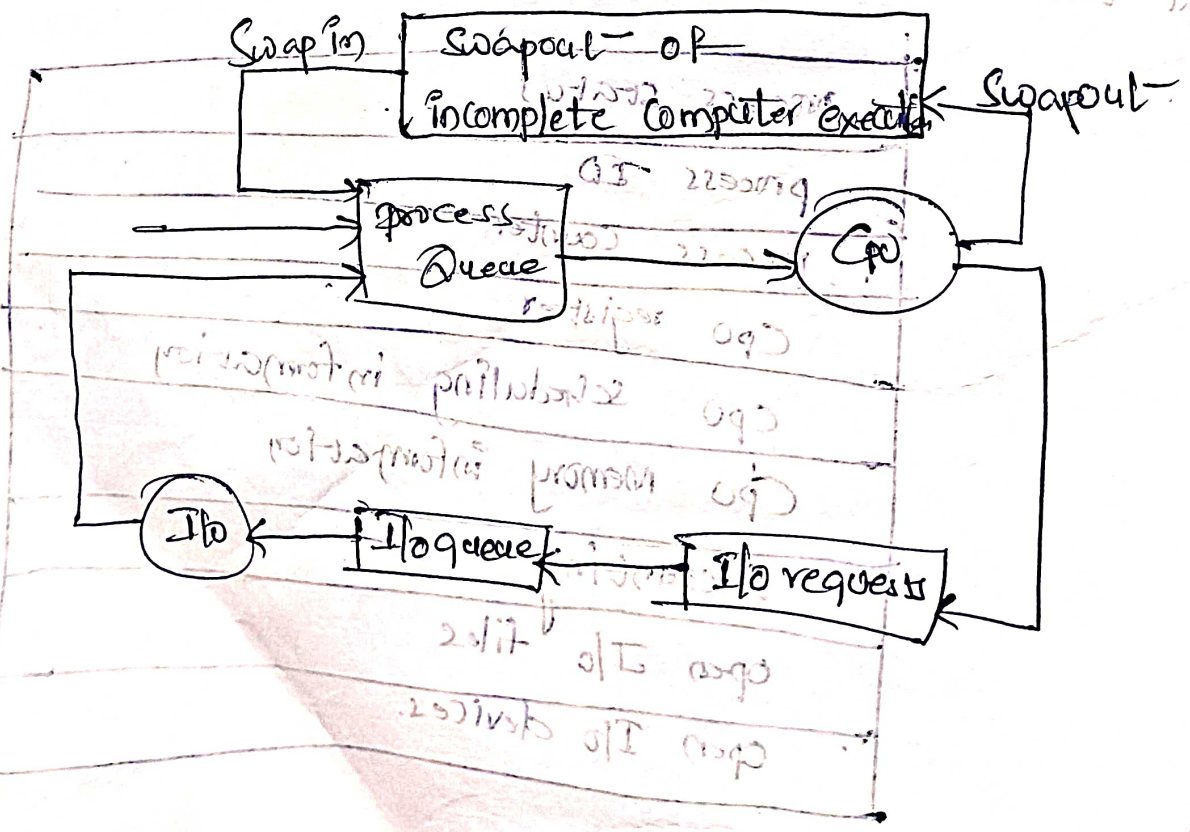
Every process has its own process control blocks. This block has different types of data that all are related to particular process



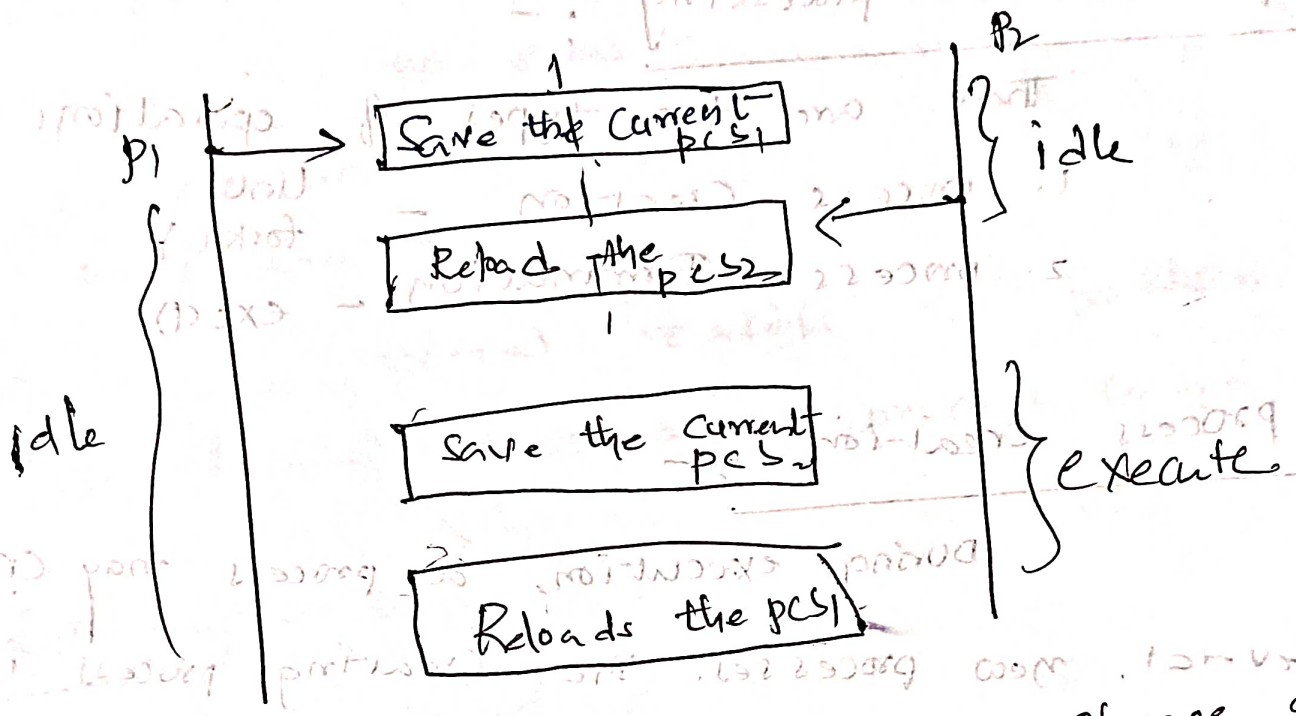
Scheduling Queues



Schedulars



Context switching :-

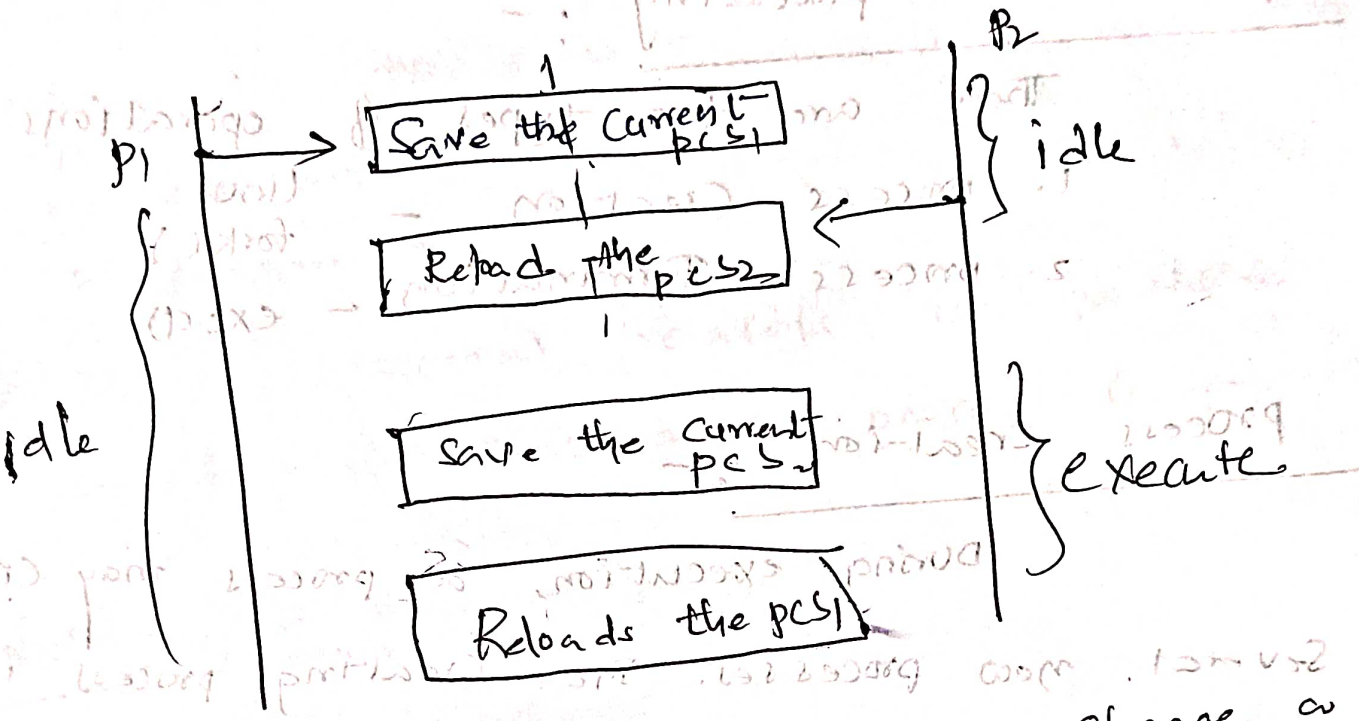


Interrupts cause the OS to change a CPU from its current task and run a kernel routine - when an interrupt occurs, the system needs to save the current context of the process running on the CPU so that it can restore that context when its processing is done, essentially suspending the process and then resuming it.

The context is represented in the PCB of the process which includes the value of CPU register, the process state etc.,

Generally we perform a state save of current state of CPU, be in Kernel (8) User Mode and they state restore to resume Operations.

Context switching :-



Interrupts cause the OS to change a CPU from its current task and run a kernel routine - when an interrupt occurs, the system needs to save the current context of the process running on the CPU so that it can restore that context when its processing is done, essentially suspending the process and then resuming it.

The context is represented in the PCB of the process which includes the value of CPU register, the process state etc.,

Generally we perform a state save of current state of CPU, be in kernel (or) user mode and they state restore to resume operations.

~~Process Operation~~ :-

Operation on processing :-

There are two types of operations
Linux
1. process Creation — fork()
2. process Termination — exec()

process Creation :-

During execution, a process may cre
several new processes. The creating process i
Called parent process and new process are
called as child process. Each of the pro
may in turn create process, forming a tr
of process

pid < 0 (no child process)
pid = 0 (one child)
pid > 0 (> one child)

process Termination :-

Process executing Concurrently in the OS may be either independent processes (OS) co-operating processes. A proces is independent if it cannot affect (o be effected by other processes executing in the system.

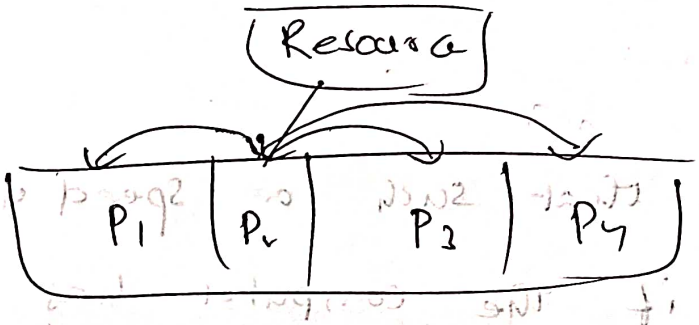
Inter process Communication :-

process are two types :-

1. Individual
2. Co-operation

There are several reasons for providing an environment that allows process co-operation

- **Information sharing :-**



Since several users may be interested in the same piece of information, we must provide an environment to allow concurrent access to such information.

When the process is co-operative effective Communication is required between parent & child process.

Modularity :-

we may want to construct the system in a modular fashion, dividing

the system functions into seperate processes (or) threads. It decomposes into small modules, subprocesses. whenever modularity occurs, inter process communication is required.

Computational speed up :-

If we want a particular task to run faster, we must break it into subtasks each of which will be executing in parallel with the others.

Notice that such a speed up can be achived only if the computer has multiple processing cores.

To speed up process one wants to Communicate others

Convenience :-

Each individual user may work on Many task. for ex :- listening mps, compiling Java program, all in same time.

The Ipc convinence and depends on two factor :- Shared memory, message

Communication in client - server environment :-

In the client - server Communication the process may go through the following programs

1. Socket :-

It is the program having the address of both client & server which is identified by a unique number with Ip address & port address there are 1024 standard port address are exists. we have to give more than 1024 as port address.

2. Rpc (Remote procedure Calls) :- The os of client & server Calls the system functions/Routines when client & server Communicating, in between there Can be another system May execute its own os Calls.

The following six steps will exists.

Step 1 :- At client side stub (or) proxy call is executed. which means the parameters will be converted into a message Called mar

Step 2 :- client send request by calling Send system Call

Step 3 :- client Can Call Recieve methods/fun Periodically to obtain the response from ser

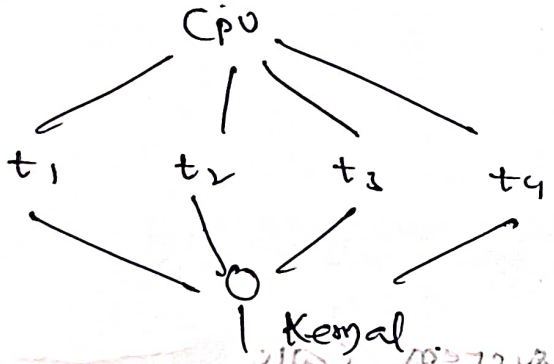
Step 4 :- At server stub (S) : Server proxy is invoked Called - it gathers client information

Step 5 :- Server OS calls a system call to divide the information into parameters & message.

Step 6 :- Server OS calls exec() method to execute the process.

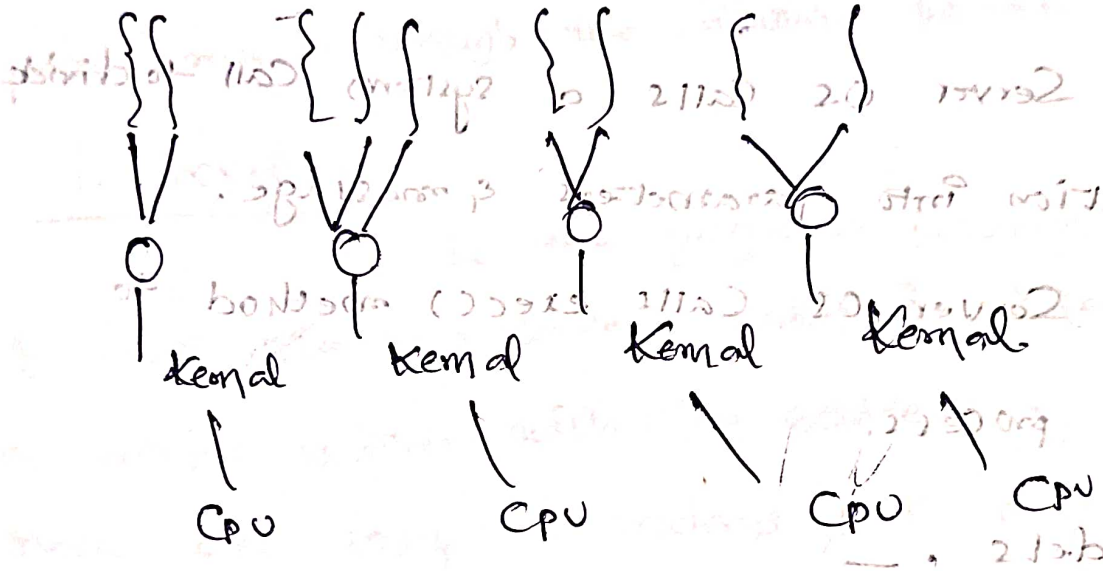
Thread models :-

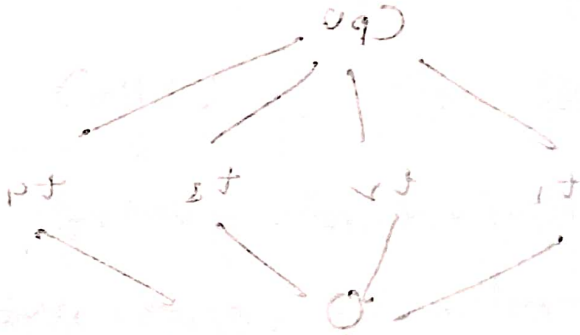
Many - to - many :-



one - to - one :-

Many to Many :-





Thread issues :-

1 → fork() 2. exec() system Calls

2. Signal Handling
3. Thread Cancellation
4. Thread local Storage
5. Thread acaration s.

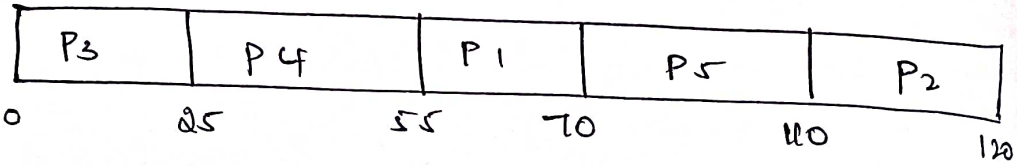
* To calculate avarage waiting Time Using Gant chart. of following

PID	B.T
P_1	15
P_2	10
P_3	25
P_4	20
P_5	40

order :- $P_3\ P_4\ P_1\ P_5\ P_2$

$P_2\ P_4\ P_5\ P_1\ P_3$

Sol :-



Waiting Time of

$P_3 \rightarrow 0$

$P_4 \rightarrow 25$

$P_1 \rightarrow 55$

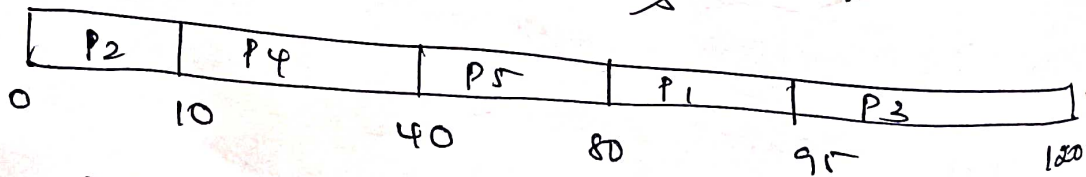
$P_5 \rightarrow 70$

$P_2 \rightarrow 110$

avarage waiting Time $= \frac{0+25+55+70+110}{5}$

$= \frac{260}{5} = 52.$

(ii)



Waiting Time of

$P_2 \rightarrow 0$, $P_4 \rightarrow 10$, $P_5 \rightarrow 40$, $P_1 \rightarrow 80$, $P_3 \rightarrow 95$

avarage waiting Time $= \frac{0+10+40+80+95}{5}$

$= \frac{225}{5}$

Shortest first job scheduling (SJF) :-

This algorithm allocates the cpu to the process which has lowest burst (execution) time

It will be either preamtive (or) non-preamtive.

In non-preamtive / cooperative, the process must be completed its execution & no process will be interupt the current execution process &

But in preamtive the current execution process may go to waiting state and new process where the burst time is the less than the executing process burst time will allocate the cpu.

In preamtive the algorithm works on

Short-next Remaining process first

SJF Non-premtive :-

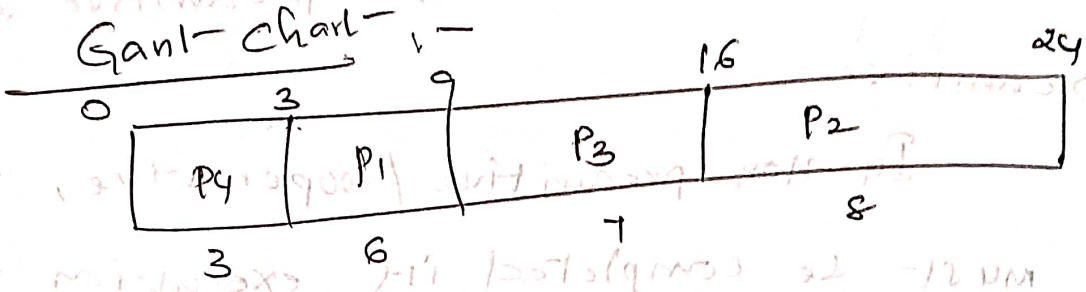
Construct the following 4 process which are arrived at same time where the burst time is notice d. calculate the average waiting time

P1 P	Burst time
P1	6
P2	8
P3	7
P4	3

Gantt chart :-

Sol :- To calculate average waiting time we should form the Gant- chart -

Gant- chart :-



Waiting time for

$P_1 = 3$ ms

Waiting time for $P_2 = 16$ ms

Waiting time for $P_3 = 9$ ms

Waiting time for $P_4 = 0$ ms

$$\text{Average waiting time} = \frac{\text{Sum of individual}}{\text{No. of processes.}}$$

$$= \frac{3+16+9+0}{4} = 28/4 = 7 \text{ ms}$$

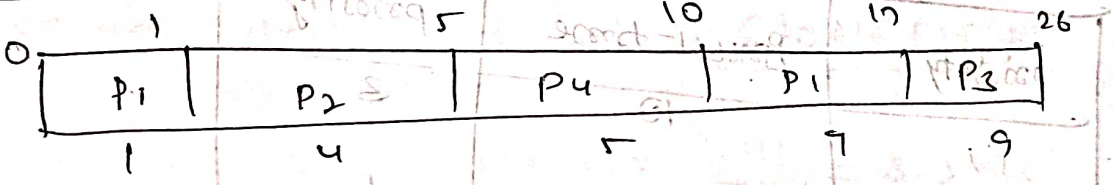
SJF premtive :-

PID	Arrival Time	Brust Time
P_1	0	8
P_2	1	4
P_3	2	9
P_4	3	5

Sol :—

PID	Arrival Time	Burst Time
P_1	0	8 − 1 = 7
P_2	1	4 − 1 = 3
P_3	2	9
P_4	3	5

Gant- chart :—



Waiting time = Total time − No. of preemptive − arrival time

Waiting time for P_1 = 10 − 1 − 0 = 9

P_2 = 1 − 0 − 1 = 0

P_3 = 17 − 0 − 2 = 15

P_4 = 5 − 0 − 3 = 2

Average waiting Time = $\frac{\text{sum of individual WT}}{\text{No. of process}}$

$= \frac{9+0+15+2}{4} = \frac{26}{4} = 6.50$

Priority :—

→ Another algorithm for process Scheduling where the process allocate the CPU based on processes priority

→ Each processes have a ~~feature~~ priority value.

→ This algorithm may have different approach like the lowest value have high priority (or) The highest value can be the high priority

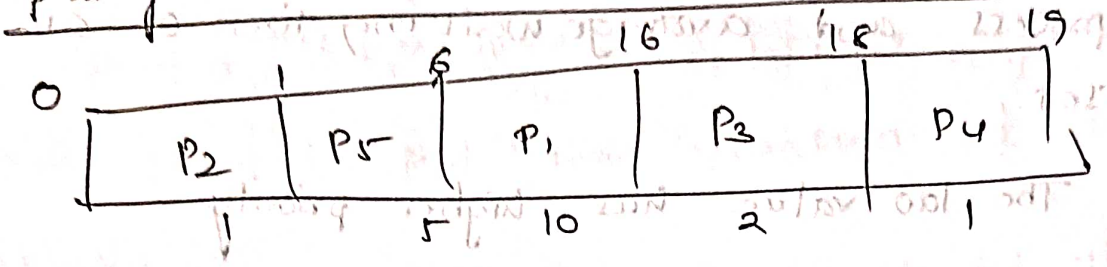
priority	Burst time	priority
P_1	10	3
P_2	1	1
P_3	2	4
P_4	1	5
P_5	5	2

→ This may be premtive (or) non-premtive.

→ In non-premtive the process can't be premted till the process is completes it's execution

→ In premtive process with high priority will allocate CPU then the executed proces priority

Priority schedule – Non-premitive (LUHP) :-



Waiting time for $P_1 = 6$

$P_2 = 0$

$P_3 = 16$

$P_4 = 18$

$P_5 = 1$

Average waiting Time $= \frac{6+0+16+18+1}{5}$

$= \frac{41}{5} = 8.2$ ms.

Priority scheduling – premitive :-

Each process may have different arrival times and priorites.

When one process is executing it can be ~~pre-entered~~ preempted for higher priority process. and the current execution process will be in the dequeue.

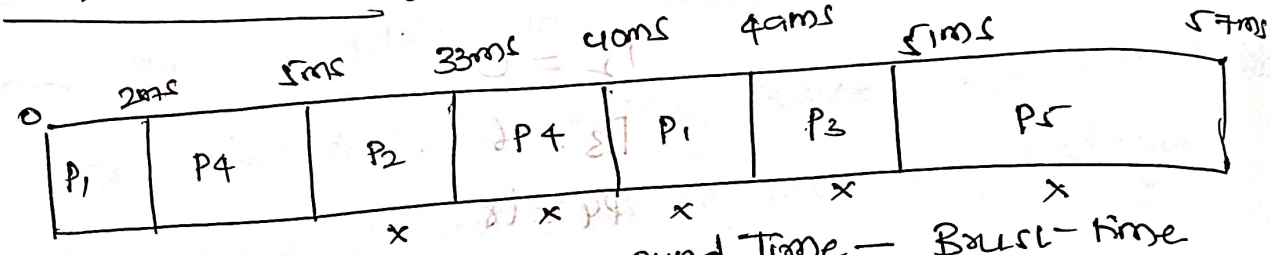
Consider the following processes with their arrival time, burst time and priority (low value is high priority)

process ID	Arrival Time	Burst Time	priority
P_1	0	11	2
P_2	5	28	0
P_3	12	2	3
P_4	2	10	1

Calculate turn around time, waiting time of each process and average waiting time of all processes.

Sol^n — The low value has higher priority

Gantt chart —



Waiting Time = Turnaround Time — Burst time

Turn around Time = Complete — AT

Turn around Time of P1 = 49 − 0 = 49

Turn around Time of P2 = 33 − 5 = 28

Turn around Time of P3 = 51 − 12 = 39

Turn around time of P4 = 40 − 2 = 38

Turn around Time of P5 = 57 − 9 = 68

Waiting time for P1 = 49 − 11 = 38

Waiting time for P2 = 28 − 28 = 0

Waiting time for P3 = 39 − 2 = 37

Waiting time for P4 = 38 − 10 = 28

Waiting time for P5 = 48 − 6 = 42

Average waiting time = $\frac{28+0+37+28+42}{5}$

$= \frac{145}{5} = 29$

Round Robin :-

It is similar to FCFS algorithm where premtion and time quantum of CPU is added.

This algorithm is applicable for time sharing operating systems.

In this algorithm cpu gives a time slice (or) time quantum for each process.

When burst time < Time @quantum, the process itself terminates and cpu allocation gives to the next process.

When burst time > Time Quantum, the premted process is added at tail of the ready queue.

When premted process is adding to the queue at the same time new process is arrived they at tail new process is added.

Process ID	Burst time
P_1	24
P_2	3
P_3	3

TAT = Complete - AT

Waiting time = TAT - BT

TAT of P_1 = 30 - 0 = 30
P_2 = 7 - 0 = 7
P_3 = 10 - 0 = 10

WT P_1 = 30 - 24 = 6
P_2 = 7 - 3 = 4
P_3 = 10 - 3 = 7

avg = $\frac{6+4+7}{3}$

= $17/3$ = 5.6667

Waiting time for each process

= last-started process - Arrival Time - (No. of premted x TQ)

WT of P_1 = 26 - 0 - (5 × 4)

= 26 - 20 = 6

WT of P_2 = 4 - 0 - 0 = 4

WT of P_3 = 7 - 0 - 0 = 7

Round Robin algorithm for different arrival time process :-

In this approach all processes are arranged as circular queue.

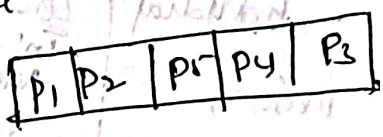
For each time quantum CPU makes an alarm, premes the executing process and allocation will be given to head of the ready queue process.

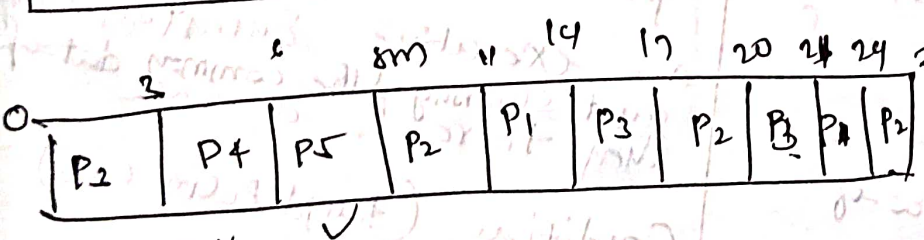
At one time (8) at the same time the queue has new process and premted process then head position is occupied by new process behind that premted process.

* Consider the following process with their arrival times and burst times. The time quantum of the CPU is 2ms. Calculate the average waiting time of all the processes

Process ID	Arrival time	Burst Time
P1	10	5
P2	1	3
P3	2	1
P4	3	2
P5	4	3

Process ID	Arrival Time	Burst Time
P1	5	6-3=3
P2	3	5-3=2
P3	0	4-3=1
P4	1	3
P5	2	2





Time Quantum = 3ms

Time 0 = P2

Time 1 = P2 P4

Time 2 = P2 P4 P5

Time 3 = P1 P5 P4

Time 4 = P1 P2 P5

Time 5 = P2 P3 P1 P2 P3

Calculate turn around time, waiting time and avarage waiting time

turn around time of P1 = CMT - AT

= 24 - 5 = 19

P2 = 26 - 2 = 23

P2 = 21 - 0 = 21

P4 = 8 - 1 = 7 P5 = 6 - 2 = 4

Waiting time of process = TAT − BT

Waiting time of process $P_1 = 15 - 6 = 9$

$P_2 = 21 - 8 = 13$

$P_3 = 22 - 7 = 15$

$P_4 = 8 - 3 = 5$

$P_5 = 9 - 2 = 7$.

Average waiting time

$= \frac{9 + 13 + 15 + 5 + 7}{5}$ $\frac{13 + 15 + 9 + 2 + 9}{5}$

$= \frac{29}{5} = 9.0$

→ It is used to waiting an resu

process Synchronization → process in wait state until i completes its execution proc

Individual

```
main()
{
 Int a
 a=a+10
}
fun()
{
 Int a
}
```

The change in one main process will not affect the

Co-operation

```
Int a;
main
{
 a=10
 a=a+10
 fun()
}
fun()
{
 a=a+5
}
```

The change in main function affect the fun() value

Critical Section :-

If two process are executing parallelling there is no problem. But the process are executing simultaneously and sharing the common data then there is a race Condition. (Round Robin case) In this case inconsistency will occur.