Page No.
Date : 25/7/23

UNIT-1

Operating System (BTCS303)

1) Concept of Operating System

Software : Collection of program is called software.
A program is collection of instructions.
Ex : Word, Excel, Office, Powerpoint, Firefox, chrome

Hardware : Physical devices in a collection of computer system is called hardware
Ex : Processors (CPU), I/O devices

2) Types of Software :
Application Software :
System Software
Utility Software

System Software : The software which is used to perform all type of system level task of computer.
Ex : OS, Linker, Loader, Compiler, Interpreter

Application software : Software which is created by users using the different languages for any special purpose
Ex : Library Management Sys, Banking Sys, Rail Res. Sys.

Utility Software : The software which provide an additional meaning to the computer system.
Ex : Calculator, Calendar, Paint, Notepad, Media players

Qus: Diff b/w multiprograming & multitasking

Page No.
Date :

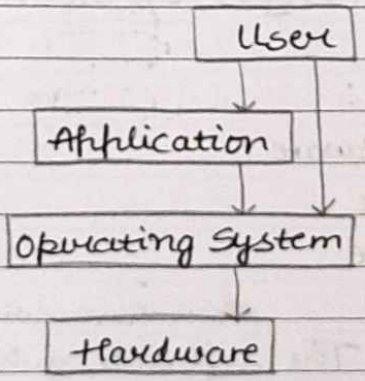
I T I W

Operating System

1. An operating system is a collection of system software that manage computer hardware resources and provides common services for the computer programs

OR

2. A program that acts as a interface / mediator b/w a user and a computer hardware.



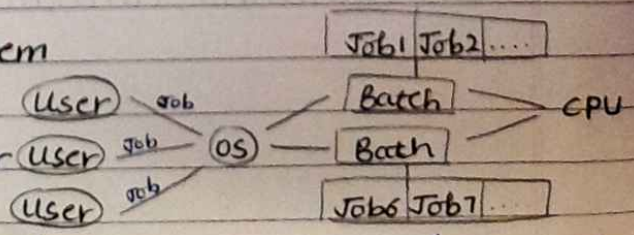
3) Goals of Operating system

1. Simplify the executions of the program and make solving problem easier
2. Use computer hardware efficiently (CPU)
3. Allow sharing of hardware and software resources
4. Improve overall system reliability (fault tolerance)
5. Make application more flexible

4) Types of Operating System

1) Batch OS :

- No direct interaction b/w user and computer
- The user has to submit a job written on the card or tape to the computer operator

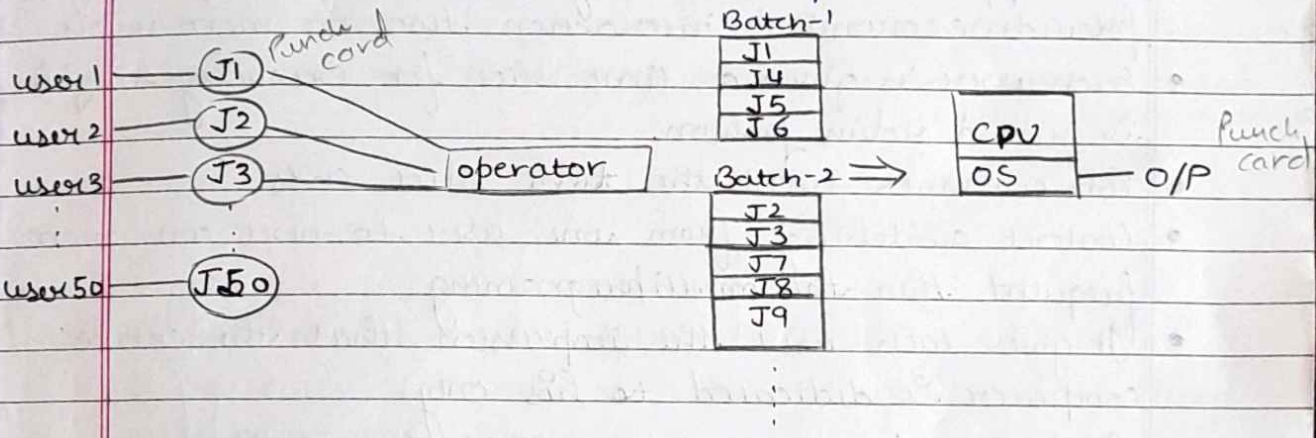


punch

Starvation: Any program waiting for longer duration for CPU.
Preemption: Switching of CPU's

Page No.
Date :

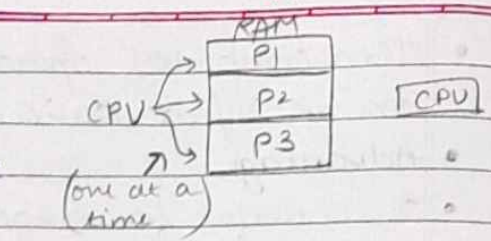
- Then computer operator places a batch of several jobs on an input device/output device
- Advantage:
- It is useful for large batches
- Disadvantage:
- No mechanism to prioritize the processes



2) Multiprogramming OS: (maximum utilisation of CPU)

- Several process are kept in main memory at the same time, the CPU is multiplexed among them
- Multiprogramming increase CPU utilisation.
- Multiple jobs are loaded into main memory and one is selected from them for the execution by CPU.
- If at some point program in progress requires services of a peripheral device (I/O devices), the control of the CPU is given to the next job which is in the main memory.
- So CPU is always executing some program instead of waiting.
- Advantage:
- High CPU utilisation if there are jobs available in RAM.
- Many programs are alloted CPU simultaneously.
- Better I/O memory utilisation.

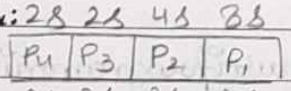
- Disadvantages:
- CPU scheduling is require
- Memory management is require



3) Multitasking OS (Time sharing system)

- It supports CPU scheduling and multiprograming to provide economical interaction two or more users.
- Each user is given a time slice for executing his job in round robin fashion.
- Job continues until the time slice ends.
- Context switching from one user to next are more frequent then the multiprograming
- It gives each user the impression that the entire computer is dedicated to his only
- Advantage:
- Provide high responsiveness
- It reduce CPU ideal time
- Disadvantages:
- If lots of users and apps are running at the same time it may be hang up the system

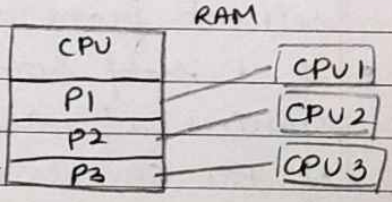
require time: 2s 2s 4s 8s



CPU = 2s

4) Multiprocessor OS: (Parallel system)

- More than one CPU works in parallel known as multiprocessor system
- Processors share memory and i/o devices, system bus and communication usually takes place through the shared memory.
  (This system is called tightly coupled system)
- Multiprocessors delivers high computing power and speed.
- All the processes executed under single OS



- Advantages:
- Increased through put (No. of process executed per unit time)
- Through put increase as there are more no. of processors
- Economical: Buying one system with 3 CPU is cheaper than buying 3 different system with 3 CPU's
- Increased reliability: The failure of 1 processor will not stop the system, it functions with other available processors
- Disadvantages: Complex, require big memory, system crash if processor in charge of task fail

5) Real Time Operating System: (RTOS)

- It promises a certain capability within a specified time constraint.
- It has two types:

1) Hard RTOS
2) Soft RTOS

HARD RTOS

* It guarantees that the maximum time for critical operations and complete them on ~~them~~ time are refer to as Hard RTOS
* If system fails to meet the deadline even once the system is considered to have fail.
* Ex: Defence Application, Missile System, Nuclear System

SOFT RTOS

* The critical task will get the priority over other task but no assurity of completing it in a dfined time
* Ex: Audio-Video streaming

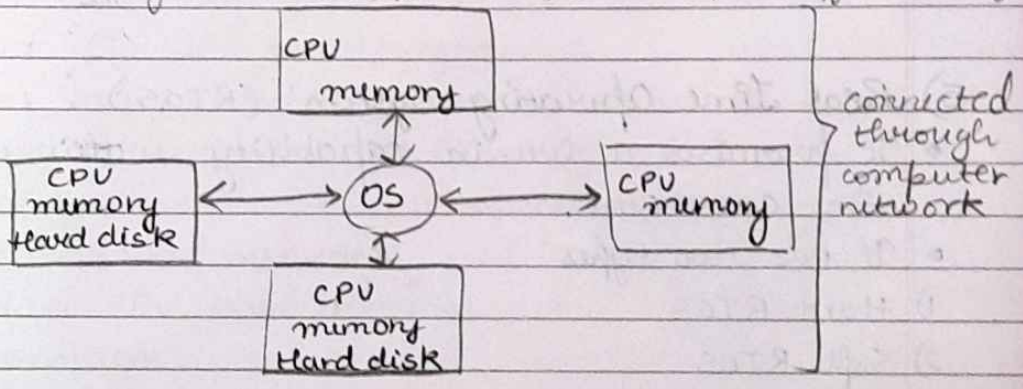
Disadvantages: Costly, Complex, Limited tasks

Page No.
Date :

- Advantages: Easy to use, Less Space, Error free

6) Distributed Operating System:

Dis: Security issue, Cost ↑, Lost data, Overload prob, diff to manage



- Advantages: Data sharing, Extent easily, Reliability, less delay, effective

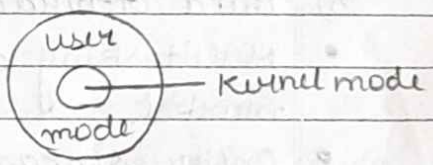
(*) Operating System Services:

1) User Interface (By GUI, CLI, CUI)
2) Program Execution (Program → memory → CPU → execute)
3) I/O operation
4) File system manipulation
5) Communication

6) Error detection
7) Resource allocation
8) Accounting (nazar rkhna)
9) Protection and security (matlabi)

1/8/23

(*) System Call
It is a request made by the user program in order to get the services of an operating system.
1) User mode (user can use only software)
2) Kernel mode (user can use hardware) (heart of OS)

→ User mode:


- When CPU is in user mode the programs do not have direct access to memory & hardware resources.
- If any program crashes only that particular program is halted, that means the system will be in a safe state. Hence, most programs in an operating system run in user mode.

→ Kernel mode:
- When CPU is in Kernel mode the programs execution can accessed any memory address and any hardware. Hence, kernel mode is a very privileged and powerful mode.
- If a program crashes in a kernel mode the entire system will be halted.

(*) Evolution of Operating System
1) First generation (1945-1955)
- No operating system
- Hardware like vaccum tube and plugboard were used

- Consume more electricity & generate lots of heat
- Need to work with machine language

Eg: ENIAC

2) Second-Generation (1955-1965)

- Batch system was introduce
- Transistor were used in place of vacuum tubes that consume less electricity & generate less heat
- Foundation for multitasking

3) Third Generation (1965-1980)

- Multitasking and multiprograming OS was in the market.
- Online storage for the system programme and the user programme was introduced
- At the place of transistor IC's were used
- No starvation

4) Farth Generation (1980 - till now)

- No starvation
- Maximum CPU utilisation
- At the place of IC's, LSIC's is used
- GUI, CLI, CUI is used
- It provides priority based execution
- Distributed OS

Eg:
- Personal computers (make a note)

Desktop
Workstation
Laptop
Notebook
Tablet
Smartphone

Page No.
Date : 3/8/23

(*) Operating System Structure

1) Simple Structure :

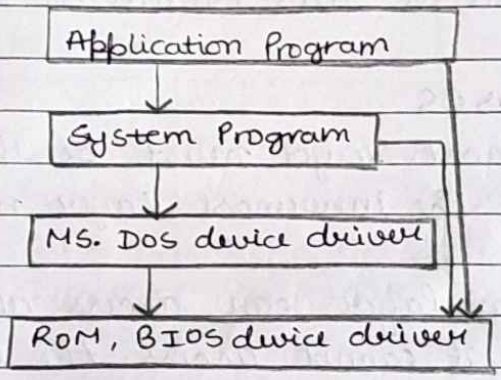


Fig: MS. DOS

Microsoft Disc OS

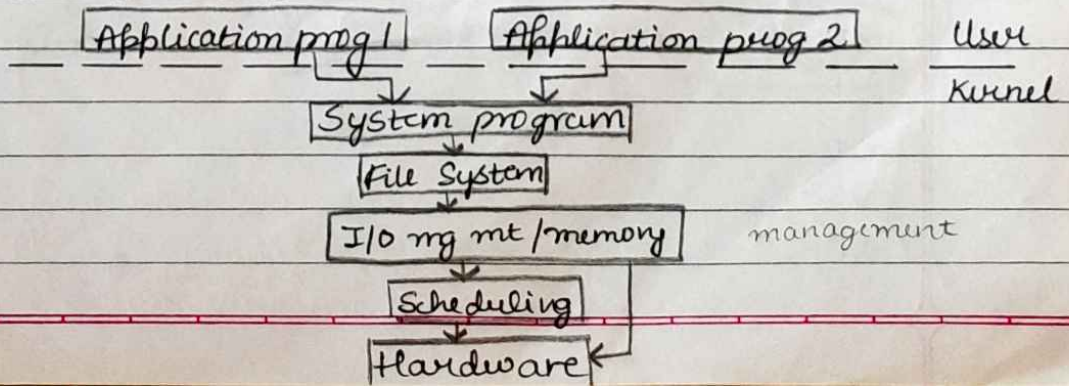
Basic I/O System

Simple Structure based operating system do not have well defined structure and small, simple and limited The interfaces ~~at the~~ and the level of functionalities are not well separated. MS. DOS is an example of such OS. In MS. DOS application programs are able to access the basic I/O routines or I/O services.
These type of OS cause the entire system to crash if one of the user program fails

Advantages : • It is easy to implement for developers.

Disadvantage : • The structure is very complicated due to no clear boundaries exist between modules.
• No Hardware protection.

3) Layered Structure :



It is a type of system in which the different services of the operating system are split into various layers, where each layer has a specific well defined task to perform and it was created to improve the simple structure

Ex: Windows OS

The outermost layer must be the user interface layer and the innermost layer must be the hardware layer.

A particular layer can access all the layers present below but it cannot access the layers present above.

Thus if the user layer wants to interact with a hardware layer then the response will be travelled through all the layer.

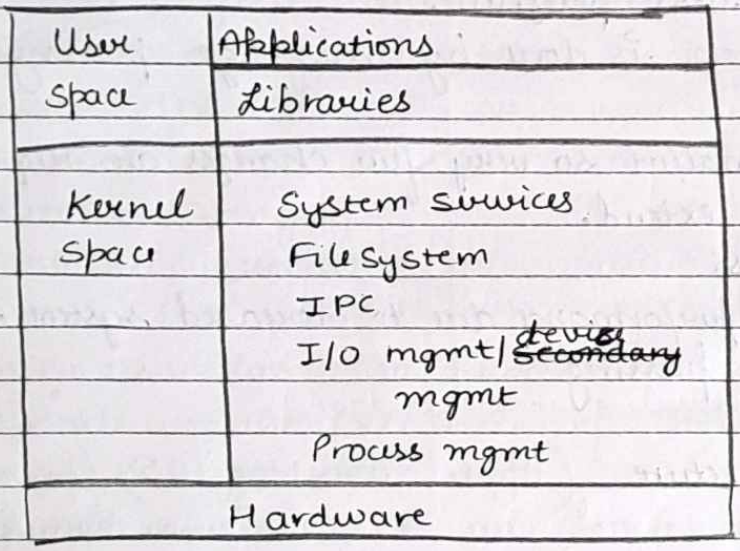
Advantages:
- Easy to update.
- Modularity
- Easy to debug.
- No direct access to the hardware
- Abstraction

Disadvantages:
- As a layer can access the services of the layers below so the arrangement of the layers must be done carefully
- Slower in execution: Layer wants to interact with another layer it sents a request that has to travel through all the layers present in between.

IPC : Inter process communication
mm mgmt : main memory management

Page No.
Date : 7/8/23

2) Monolithic architecture :

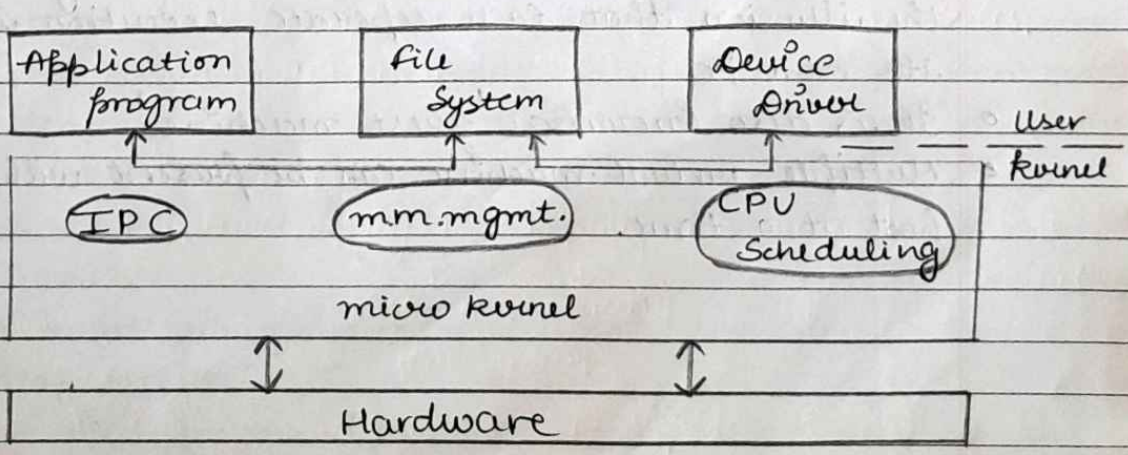


The functionality of the operating system is activated with simple function call within the kernel. In monolithic the kernel size is very large.

Advantages : • The kernel is faster due to that the execution of the processes is also faster.

Disadvantages : • The kernel size is too large if any services fail in the kernel it leads to fail the entire system

4) Micro kernel Structure :



- In the micro kernel structure the OS, removing all unneccesary part of the kernel and implement them at the user level program.
- Communication is done by message passing

Advantages:

- Kernel is smaller so very few changes are required in it
- OS can easily extend.

Disadvantages:

- It has poor performance due to increased system overhead of message passing

5) Hybrid Structure (Apple iOS, Mac OS) (Layered sys)

(interface) Application Layer
(graphics, video) Media Layer
(functionality) Core Services
(iOS technology) Core OS

microkernel
Interface
Applications
Graphics
Core OS

(*) Virtual Machine

- Virtual machine does abstract hardware of a single computer like CPU, memory, disc drivers etc, for the several executions and thereby it create the illusion that each seperate execution is running its own PC
- It is also known as 'guest machine'
- Multiple virtual machine can be present within a single host at a time.

Resource: Memory, I/O devices, hardware

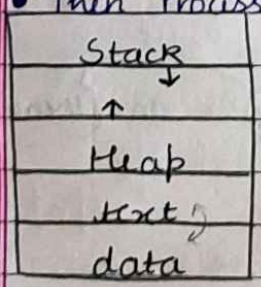
Page No.
Date: 8/8/23

Unit-2

(*) **Process:** Program under execution is called process
There are many attributes associated with process like
1) process ID
2) Process State
3) CPU Scheduling
4) Resource allocation

(*) OS goals related to process
1) Interleave the execution of Processes to maximize the CPU utilization
2) Allocate resources to the Process.
3) It supports interprocess communication (synchronization)

NOTE:-
- Programs are executable file generally stored in secondary memory.
- In order to execute program it must be loaded in main memory.
- When a program loaded into the main memory it becomes Process.
- Then Process memory can be divided into four parts:



1) Text Section:
It is made up of compiled program code, read from the secondary storage when the program is launched

2) Data section:
It is made up of global and static variable.

3) Heap section:
It is used for the dynamic memory allocation and is managed by new, delete, calloc, malloc etc.

Memory
Physical Logical

Page No.
Date :

4) Stack section:
It contains the temporary data such as functions, return address and local variable.

Imp

Program	Process
1. A set of instruction is program	A program under execution is process
2. A program is a passive entity	A process is active entity.
3. A program has longer lifespan	A process has a limited timespan
4. A program is stored on the disk in some file, it does not contain any resource	Process contains various resources like memory address, disk, printer etc as per the requirement.
5. A program requires memory space to store all the instructions	A process contains memory address which is known as 'address space'

1. Process ID:
Every process has unique ID

2. Process State:
As a process execute it changes state

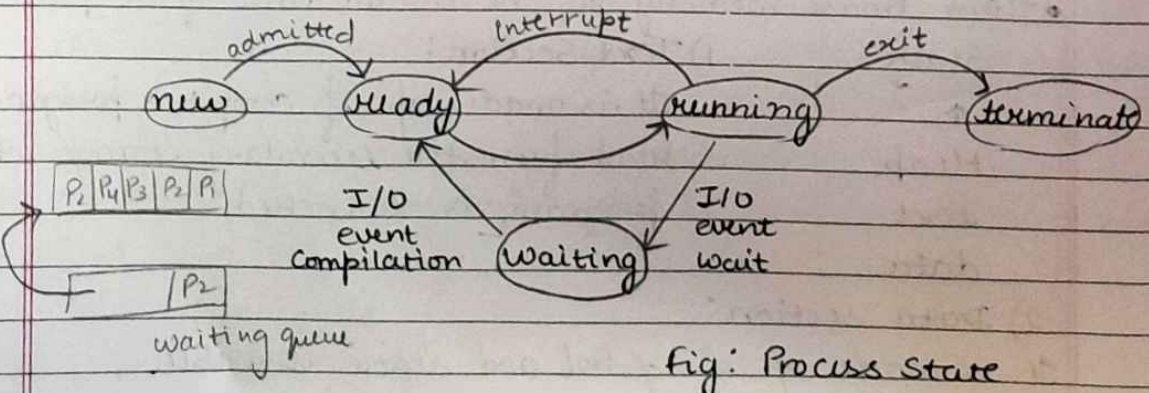


fig: Process State

new: The process is being created is called the new state
ready: The process is waiting in ready queue to be assigned to a processor.
Ready processes are waiting to have CPU allocate to them by OS, so that they can run and process may come into this state after new state or while running but interrupted by the scheduler to assign CPU some other process or after I/O completion.

9/8/23

Process Transition:

- Ready → running: It is a time the dispatcher (scheduler) selects a new process from the ready state to running state.
- Running → Ready: The running process has expired his time slot, the running process gets interrupted because a higher priority process is in the ready state.
- Running → Blocked: When a process request for something for which it must wait a service that the OS is not ready to perform an access to a resource not available, initiates I/O and must wait for the result for a process to provide input
- Blocked → Ready: When the I/O operation has finished then scheduler remove the process from the waiting (blocked) state and put it into the ready state.

Process Suspension:
when the process in the main memory is entered into the block state, the OS suspend one process by putting it in the suspend state and transfer to the disk.
When none of the process occupying in ready state, OS swap ~~block~~ suspended process into ready queue.

NOTE: PCB is maintain for a process throughout its lifetime and is deleted by OS once the process terminate.

Page No.
Date: 10/8/23

(*) Process Control Block

Process ID
State
Pointer
Priority
Program Counter
CPU register
I/O information
Accounting info
etc.

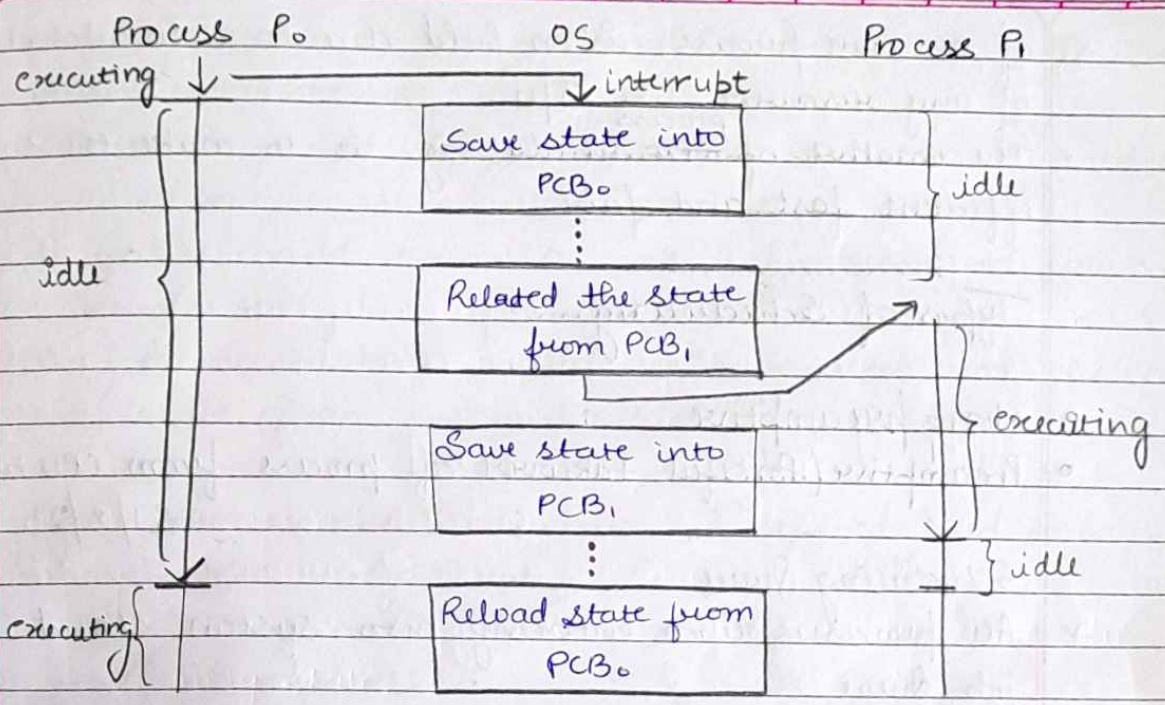
- A process control block is a datastructure that maintained by the OS for every process
- The PCB is identified by the process ID i.e. PID.
- It keeps all the information needed to keep track of all the processes.

1) PID: Every process has unique ID
2) Process State: This field maintain the current state of the process. Ex whether it is in ready state, running wait or whatever.
3) Pointer: A pointer points to the parent process.
4) Priority or privilege: This field is required to allow or disallow to access the resources
5) Program Counter: PC is a pointer to address of the next instruction to be executed for particular process
6) CPU register: It stores the information about the process those who are running currently
7) I/O information: It gives the status of I/O devices whether it is allocated or not.
8) Accounting information: It includes the amount of CPU used for the process execution.

17/8/23

(*) Context Switching

A context switching is a procedure that a computer CPU follows to change from one process to another process



CPU switching from Process$_0$ to Process$_1$

Context switching is a mechanism to store and restore the state or context of a CPU in Process Control Block (PCB). So, that a process execution can be resumed from the same point at a later time. Using this technique a context switcher enables multiple process to share a single CPU

(*) Process Scheduling

- The act of determining which process is in ready state and should be moved into running state is known as process scheduling.
- The prime aim of process scheduling system is to keep CPU busy all the time and to deliver minimum response time for all the programs.
- The scheduler must apply appropriate rules for swapping process in and out of CPU.
- Its allow one process to use the CPU while the execution

of another process is on hold due to unavailability of any resource like I/O.

- The motive of process scheduling is to make the system efficient, fast and fair.

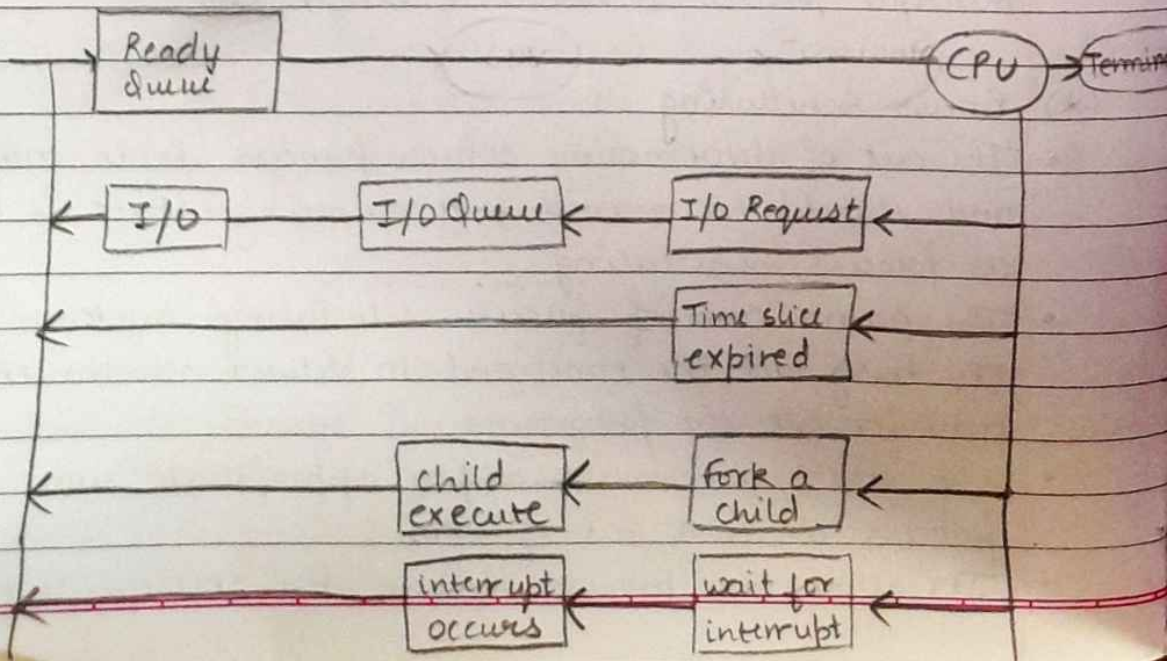
Types of Scheduling

- Non-preamptive
- Preamptive (forceful takeover of process from CPU by OS)

21/8/23

Scheduling Queue

* All processes upon entering into system stored in job Queue
* Processes in ready state are placed in the ready Queue
* Processes waiting for a device to become available are placed in device Queue or I/O Queue
* A new process is initially put in the job Queue and it waits in ready Queue until it is selected for the execution



Page No.
Date :

Once the process is assigned to CPU one of the following several events occurs:

1) The process could issue an I/O request then be placed in I/O Queue.
2) The process could create a new subprocess and wait for its termination.
3) The process could be removed of necessity from the CPU as a result of an interrupt need to be put back in the ready Queue.
4) If the process time Quantum is expired need to be put back in the ready Queue.

Types of Schedulers:

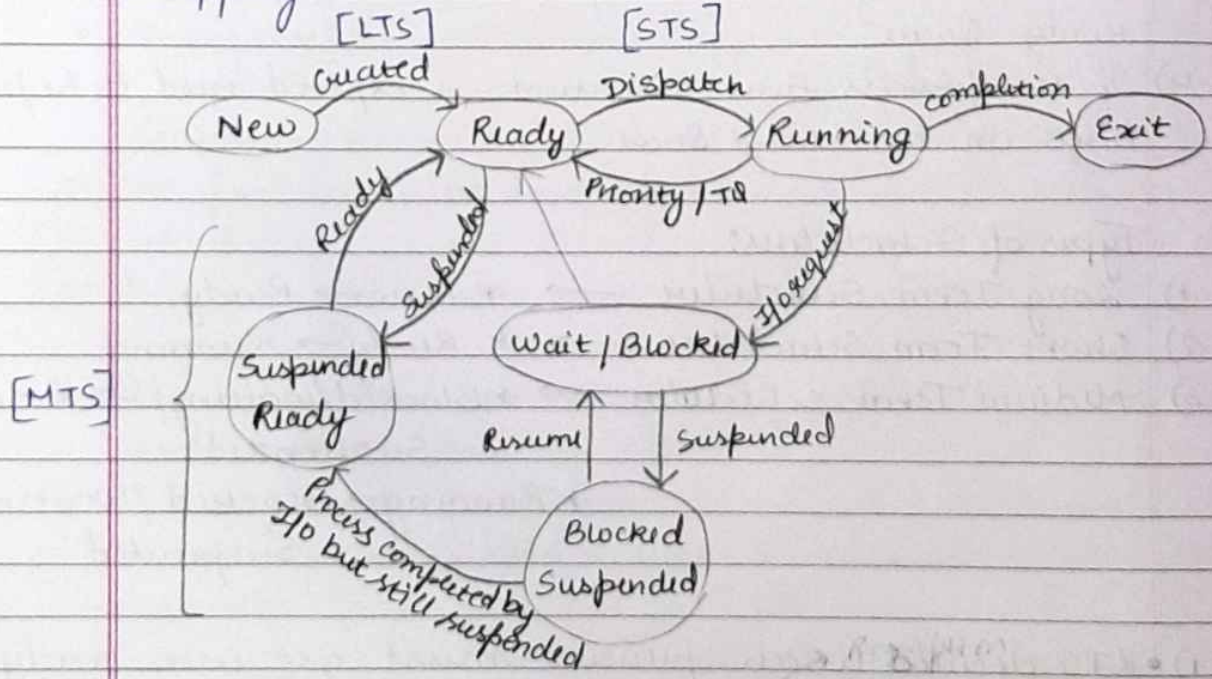
1) Long Term Scheduler → * New → Ready
2) Short Term Scheduler → * Ready → Running
3) Midium Term scheduler → * Blocked/waiting/Suspendid → Ready
   * Running → Blocked/Waiting Suspended

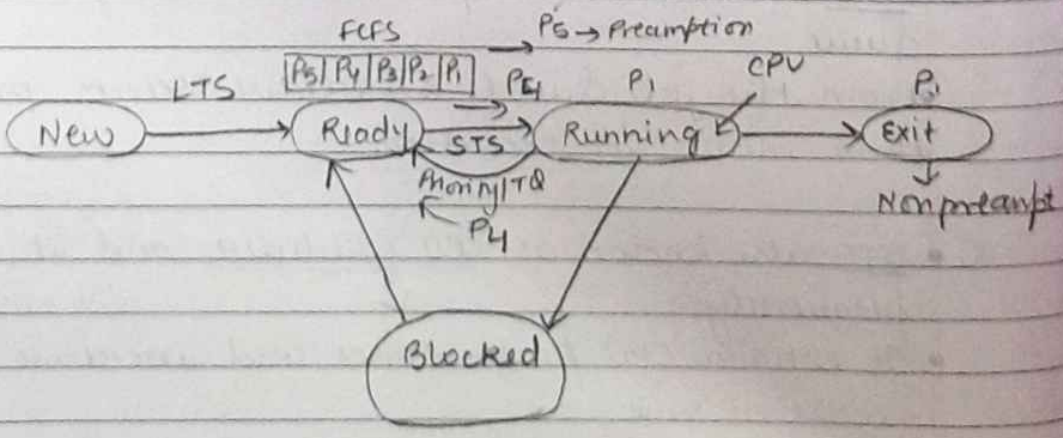
1) • LTS decides which process must get into ready Queue.
   • From the job Queue scheduler selects process and load them mem. for execution.

2) • STS also known as CPU scheduler and it runs very frequently.
   • It enhance CPU performance and increase process execution wait.
   • It slects the process from ready Queue for execution.

3)

3) MTS: During extra load this scheduler picks up big processes from ready Queue for sometime and to allow the smaller processes to execute
Thereby ~~by it~~ it reduce no of process in ready Queue
At some later time the same process can be reintroduced into main memory (ready Queue) and its execution can be continued. This scheme is called Swapping.





K♡D

Page No.
Date: 22/8/23

Scheduling Criteria

1) CPU utilization.
To make out the best use of CPU and not to waste any CPU cycle, CPU could be working most of the time. Considering the real tag system CPU usage range from 40% - 90%.

2) Throughput
Total no. of processes completed per unit time

3) Turn Around time (TAT)
The amount of time taken to execute a particular process, the interval from time of submission ~~and~~ of process to time of completion of process
TAT = Complution time - Arrival time

4) Waiting time (WT)
The sum of the periods ~~was~~ spends waiting in ready Queue, amount of time a process has been waiting in ready queue to acquire/get control on CPU

5) Load average
It is the ~~wr~~ average no. of process residing in ready queue waiting for their turn to get CPU

6) Response time
The amount of time it takes from when a request was submitted until the first response is produced

Page No.
Date :

Scheduling algorithms

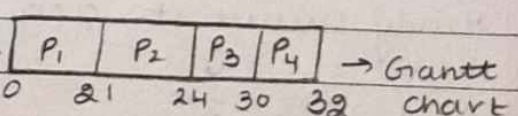
1) FCFS (first come first serve)
2) Shortest Job First (SJF)
3) Priority Scheduling
4) Round Robin Algorithm
5) Multilevel Queue scheduling
6) Multilevel feedback Queue scheduling

first come first serve scheduling (N-P)

* In this scheduling algorithm the process which arrives first, gets executed first.
* The process which request the CPU first gets the CPU allocated first.
* It is just like FIFO Queue Datastructure
* This scheduling is used in batch system.
* A perfect real time example is buying tickets from ticket counter.

Q

Process	BT	TAT	WT
P1	21	21	0
P2	3	24	21
P3	6	30	24
P4	2	32	30



→ Gantt chart

→ RQ

Calculate:
avg TAT = $\frac{107}{4}$
avg WT = $\frac{75}{4}$
= 26.75
= 18.75

NOTE: 1) It is non-preemptive algorithm which means process priority and Time Quantum (TQ) doesn't matter
2) Resource utilization is in parallel is not possible which leads to poor resource utilisation.

NOTE: 1) Arrival time :- The time in process as arrived into the ready Queue is called the Arrival time

2) Completion time :- The time when process is completed its execution is called completion time.

3) Burst time / Execution Time :- The time required for the process, for its execution is called Burst time / execution time

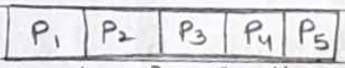
4) Average TAT :- $\frac{CT - AT}{\text{no. of process}}$

5) Average WT :- $\frac{TAT - BT}{\text{no. of process}}$

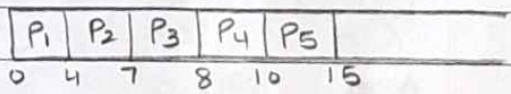
Ex:

Process	AT	BT
P_1	0	4
P_2	1	3
P_3	2	1
P_4	3	2
P_5	4	5

RQ



GC

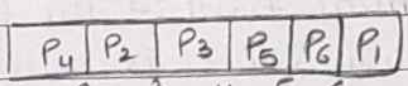


TAT	WT
4	0
6	3
6	5
7	5
11	6
$= \frac{34}{5}$	$= \frac{19}{5}$
$= 6.8$	$= 3.8$

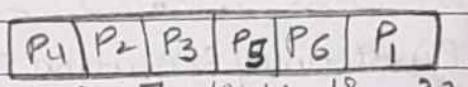
Ex:

Process	AT	BT
1	6	4
2	2	5
3	3	3
4	1	1
5	4	2
6	5	6

RQ



GC



	TAT	WT
1	③ 16	① 12
2	5	0
3	7	4
4	1	0
5	8	6
6	13	7
	50/6	29/6
	= 8.3	= 4.8

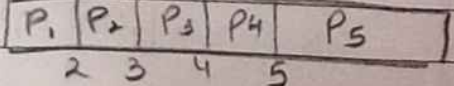
Shortest Job first (SJF) (N-P)

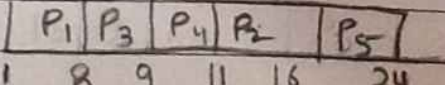
* SJF scheduling works on process with shortest burst time
* To successfully implement it the burst time of process should be known to the processor in advance which is practically not feasible all the time.
* This scheduling algorithm

Ex:

P.No.	AT	BT
P1	1	7
P2	2	5
P3	3	1
P4	4	2
P5	5	8

RQ





	TAT	WT
1	(4) 7	0
2	14	9
3	6	(2) 5
4	7	5
5	19	11
	63	30
	$= \frac{53}{5}$	$= \frac{30}{5}$
	$= 10.6$	$= 6$

28/8/23

Q

P.No.	AT	BT
1	6	1
2	3	3
3	4	6
4	1	5
5	2	2
6	5	1

P4	P5	P2	P3	P6	P1
1	2	3	4	5	6

P4	P6	P1	P5	P2	P3
1	6	7	8	10	13	19

	TAT	WT
1	(2) 2	1
2	10	7
3	15	9
4	5	0
5	8	6
6	2	1
	$= \frac{42}{6}$	$\frac{24}{6}$
	$= 7$	$= 4$

=7.17

Q Mode: Preemptive
Criteria: BT

P.No.	AT	BT	CT	TAT	WT
1	0	~~7~~ 6 0	19	②19	①12
2	1	~~5~~ 4 0	12	12	7
3	2	~~3~~ ~~2~~ 1 0	5	5	3
4	3	~~1~~ 0	6	~~2~~ ~~4~~ 3/6	1 ~~2~~ 4/6 = 4
5	4	~~2~~ 0	4	1	0
6	5	~~1~~ 0	1	~~2~~ 4/6	~~2~~ 4/6

~~P1~~	~~P2~~	P3	P4	P5	P6
0	1	2	3	4	5

P1	P2	P3	P4	P3	P6	P5	P2	P1
0	1	2	3	4	6	7	9	13, 19

Q

P.No.	AT	~~B~~ BT	CT	TAT	WT
1	3	~~4~~ 0	13	10	6
2	4	~~2~~ 0	9	5	3
3	5	~~1~~ 0	7	2	1
4	2	6	19	17	11
5	1	8 7	26	25	17
6	2	~~4~~ ~~3~~ ~~2~~ 1 0	16	4	0
				=63/6	=38/6
				=10.5	=6.3

P5	P4	P6	P1	P2	P3
1	2	3	4	5	

P5	P6	P6	P6	P6	P3	P2	P1	P4	P5
1	2	3	4	5	6	7	9	13	19	26

Page No.
Date : 29/8/23

Round Robin Schedulling

- Each process gets a small unit of CPU time (Time Quantum) usually 10-100 ms after this time has elapsed, the process is preempted and added to the end of the Ready Queue
- If there are 'n' processes in ready Queue and time Quantum is 'q' then each process gets '1/n' of CPU time in chunks of atmost 'q' times. No process waits more than '(n-1)q' time unit.
- It is preempted

Ex:

Process	AT	BT	TAT	WT	TQ=2
1	0	~~4 2 0~~	③ 8	① 4	
2	1	~~5 3 1~~ 0	17	12	
3	2	~~2~~ 0	4	2	
4	3	~~1~~ 0	6	5	
5	4	~~6 4 2~~ 0	17	11	
6	6	~~3 1~~	13	10	
			= 65/6 = 10.83	= 44/6 = 7.3	

Ready Queue: ~~P1~~ ~~P2~~ ~~P3~~ ~~P1~~ ~~P4~~ ~~P5~~ ~~P2~~ ~~P6~~ ~~P5~~ ~~P2~~ ~~P6~~ ~~P5~~

0 1 2 3 4 6

Running State: | P1 | P2 | P3 | P1 | P4 | P5 | P2 | P6 | P5 | P2 | P6 | P5 |

0 2 4 6 8 9 11 13 15 17 18 19 21

Q

P1	0	~~21~~ ~~16~~ ~~11~~ 32	11	
P2	1	~~3~~ 0	7	4
P3	2	~~6~~ 1	19	13
P4	3	~~20~~ 12	10	0

TQ=5, = 17.5 = 9.5

P1	P2	P3	P4	P1	P3	P1

0 5 8 13 15 20 21 32

Page No.
Date :

TQ = 3

Q)

Process	AT	BT	TAT	WT
1	5	~~5 2 0~~	27	22
2	4	~~6 3 0~~	23	17
3	3	~~7 4 1 0~~	30	23
4	1	~~9 6 3 0~~	29	20
5	2	~~2 0~~	4	2
6	6	~~3 0~~	15	12
			128/6	96/6
			= 21.33	= 16

$P_4\ P_5\ P_3\ P_2\ P_4\ P_1\ P_6\ P_3\ P_2$

P_4	P_5	P_3	P_2	P_4	P_1	P_6	P_3	P_2	P_4	P_1	P_3
1	4	6	9	12	15	18	21	24	27	30	32	33

TQ = 2

Q)

Process	AT	BT	TAT	WT
1	1	~~7 5 3 1 0~~	③ 19	① 12
2	2	~~5 3 1 0~~	15	10
3	3	~~1 0~~	3	2
4	4	~~2 0~~	8	6
5	5	~~8 6 4~~	19	11
			64/5	41/5
			= 12.8	= 8.2

$P_1\ P_2\ P_3\ P_1\ P_2\ P_4\ P_5$

P_1	P_2	P_3	P_1	P_2	P_4	P_5	P_1	P_2	P_5	P_1	P_5
1	3	5	6	8	10	12	14	16	17	19	20	24

4/9/23 Priority Scheduling

1) Priority is assign for each process
2) Process with highest priority is executed first and so on
3) Process with same priority are executed in FCFS manner
4) Priority can be decided based on memory requirement, time requirement and other resource requirement.
5) It is having 2 types:
   - Preemptive & Non-preemptive

Page No.
Date :

Q Mode is Non-preemptive and criteria is priority

1)

Priority	Process	AT	BT	TAT	WT
2	P_1	0	21	① 21	② 0
1	P_2	1	3	31	28
4	P_3	2	6	25	19
3	P_4	3	2	26	24
				103	71
				4	4

P_1	P_3	P_4	P_2
0	21	27	29	32

2)

Priority	Pid	AT	BT	TAT	WT
4	1	0	4	③ 4	0
5	2	1	5	15	10
7	3	2	1	3	2
2	4	3	2	15	13
1	5	4	3	17	14
6	6	5	6	5	0
				$\frac{60}{6} = 10$	$\frac{39}{6} = 6.5$

P_1	P_3	P_6	P_2	P_4	P_5
0	4	5	11	16	18	21

3) Preemptive, priority.

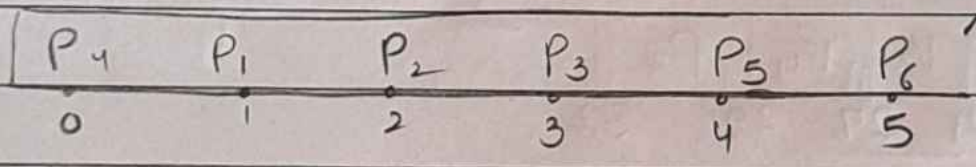
Priority	Pid	AT	BT	TAT	WT
~~4~~	1	1	~~4~~ 3 ~~2~~ 0	③ 17	① 13
~~5~~	2	2	~~2~~ 0	12	10
~~7~~	3	2	~~3~~ ~~2~~ 0	8	5
~~8~~	4	3	~~5~~ ~~4~~ 0	5	0
~~5~~	5	3	~~1~~ 0	12	11
~~6~~	6	4	~~2~~ 0	8	6
				$\frac{62}{6} = 10.3$	$\frac{45}{6} = 7.5$

P_1	P_3	P_4	P_4	P_3	P_6	P_2	P_5	P_1
1	2	3	4	8	10	12	14	15	18

Page No.
Date :

Q-

Priority	Pid	AT	BT	TAT	WT
4	1	1	4	③16	②12
× 5	2	2	2	10	8
× 7	3	3	3	5	2
× 8	4	0	~~5 4 3~~ 0	5	0
× 5	5	4	1	9	8
× 6	6	5	2	5	3
				50/6	33/6



=8.33 = 5.5