Ch - 2 Processes.

* Explain Process Control Block.

=> An Operating System supports multi-programming.

In this system multiple Process ~~Operating System~~ can run at a same time.

In this system, it needs to track all the process.

For this task, the Process Control Block track the each Process information.

Each Process have different Process Control Block.

Using Process Control Block, we can track all the details of Process.

A Process Control Block contain all the information like Process ID, Process Register, Process Number Or Process State etc.

Process ID Process State Process Counter Process Register Program Counter Process Priority List of open File General Purpose Register

Process Control Block.

- Process ID : When Process is created than every Process assigned one unique ID.

- Process State : After the create Process, Process goes in different state.

- Process Register : This Register contain all the CPU Register

- Program Counter : It stores the next instruction that is executed in Process.

- Process Priority : Process with high Priority is executed first in CPU.

- List of Open File : During the execution of every Process uses some File which need to be present in main memory.

- General Purpose Register : Every Process has its own set of register which is Used to hold the data.

* Explain Context Switching :

=> Context Switching is method used by the Operating system to switch a process from One state to other state.

Context Switch the Process and Give new Process CPU.

When Context switch is perform in system, it stores the old running process's status and assign the CPU to a new Process to execute.

Context switching helps to share a single CPU across all process.

When high priority Process falls into the queue, then Context switching stop running Process execute and assigne CPU for high priority Process execution.

If any Process require some I/o requirement then context switching stop this process execution and assign CPU for new Process execution.

If any interrupts occur while running a process in the OS then context switching stop this Process execution and assign CPU for new process execution.

A context switching allows a single CPU to handle multiple process requests.

* Explain Thread :

=> Thread is contain sequential instruction of Process.

One Process can contain multiple thread.

Each thread of the same proces makes use of PC, Stack and control blocks.

Thread is called as a lightweight process.

Thread can share comman data and do not need to use inter-process communication.

Context switching is faster when working with thread.

It takes less time to terminate a thread than a process.

- Benefits of Thread:

1 Thread can enhanced throughput of the system.

2 When you have more than one thread in system, you can schedule more than one thread in more than one process.

3 The context switching time between threads is less than a process context switching time.

4 When thread completes its execution, that process can responded very fast.

5 Multiple thread communication is simple because the thread share the same address.

6 Resources can be shared between all threads within a process.

SMVS
Page No.
Date : / /

- Types of Thread or Difference between User-level and Kernel level thread.

	User-level	Kernel-level
1	It is implemented by the user.	It is implemented by the OS.
2	Context switch time is less.	Context switch time is more.
3	Multithread is not able.	Multithread is able.
4	Easy to implement.	Complicated to implement.
5	Operating System does not recognize	Operating System is recognize.
6	It does not need hardware support	It requires harware support.
7	This thread can create and managed faster	This thread can create and managed takes time.

* Explain Multithread System in Operating System.

=> Multithreading allows the application its task divided into individuals thread.

In Multi-threads, the same process can by done by the number of threads.

Multithreading system can run multiple task at a same time.

Multithreading System can be done using User-level and Kernel level thread.

There are three posible relationship in this System.

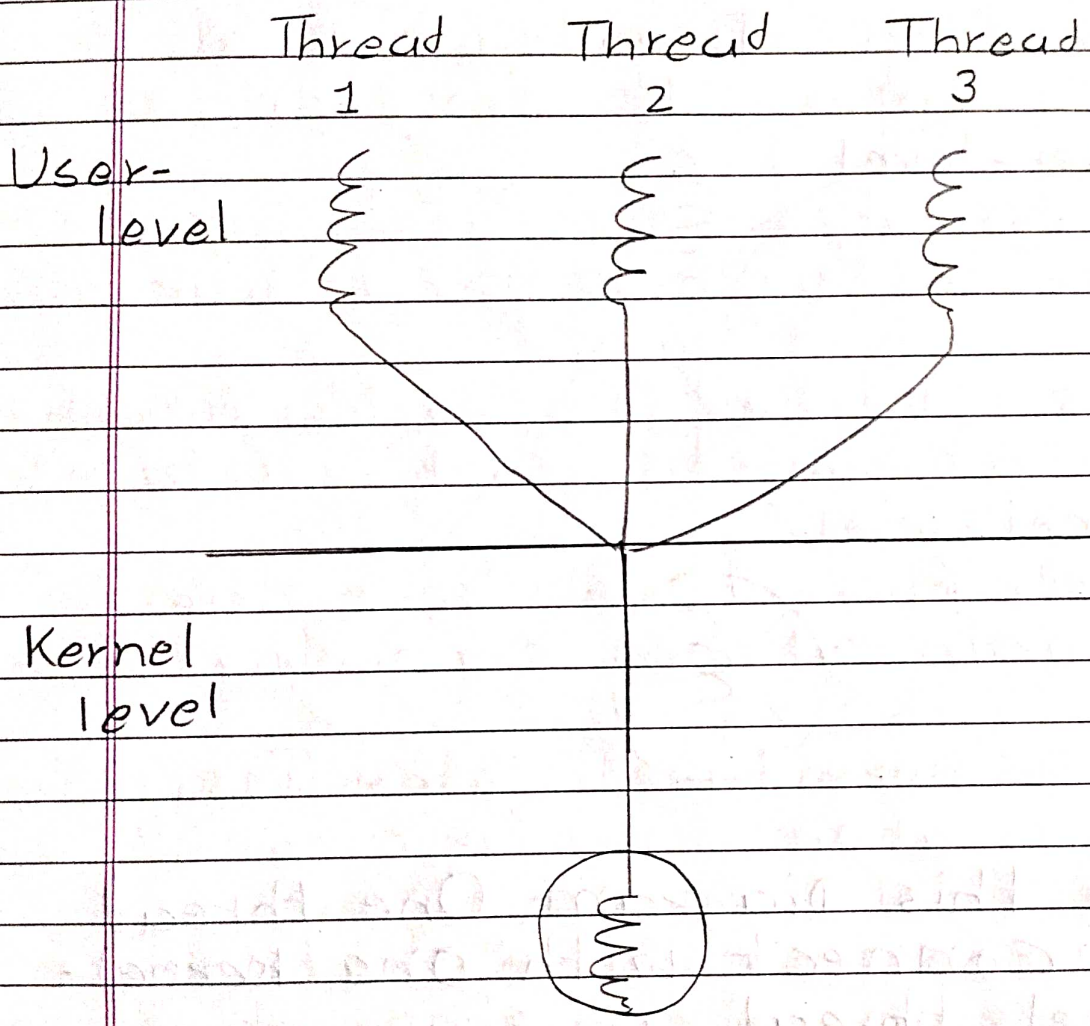
1) Many to one Multithread

2) One to one Multithread

3) Many to Many Multithread.

1 Many to one Multithreading:

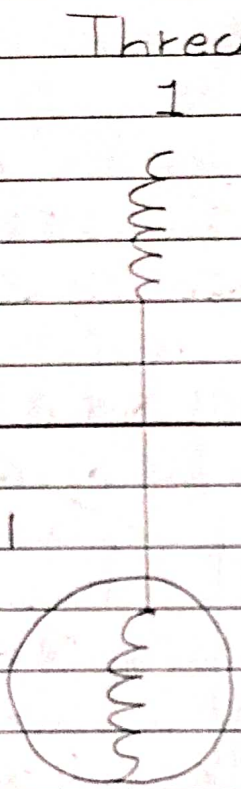
In this model many user level threads maps to one kernel-level threads.



In this diagram all the user level thread are connected with one kernel level thread.

2 One to One Multithreading:

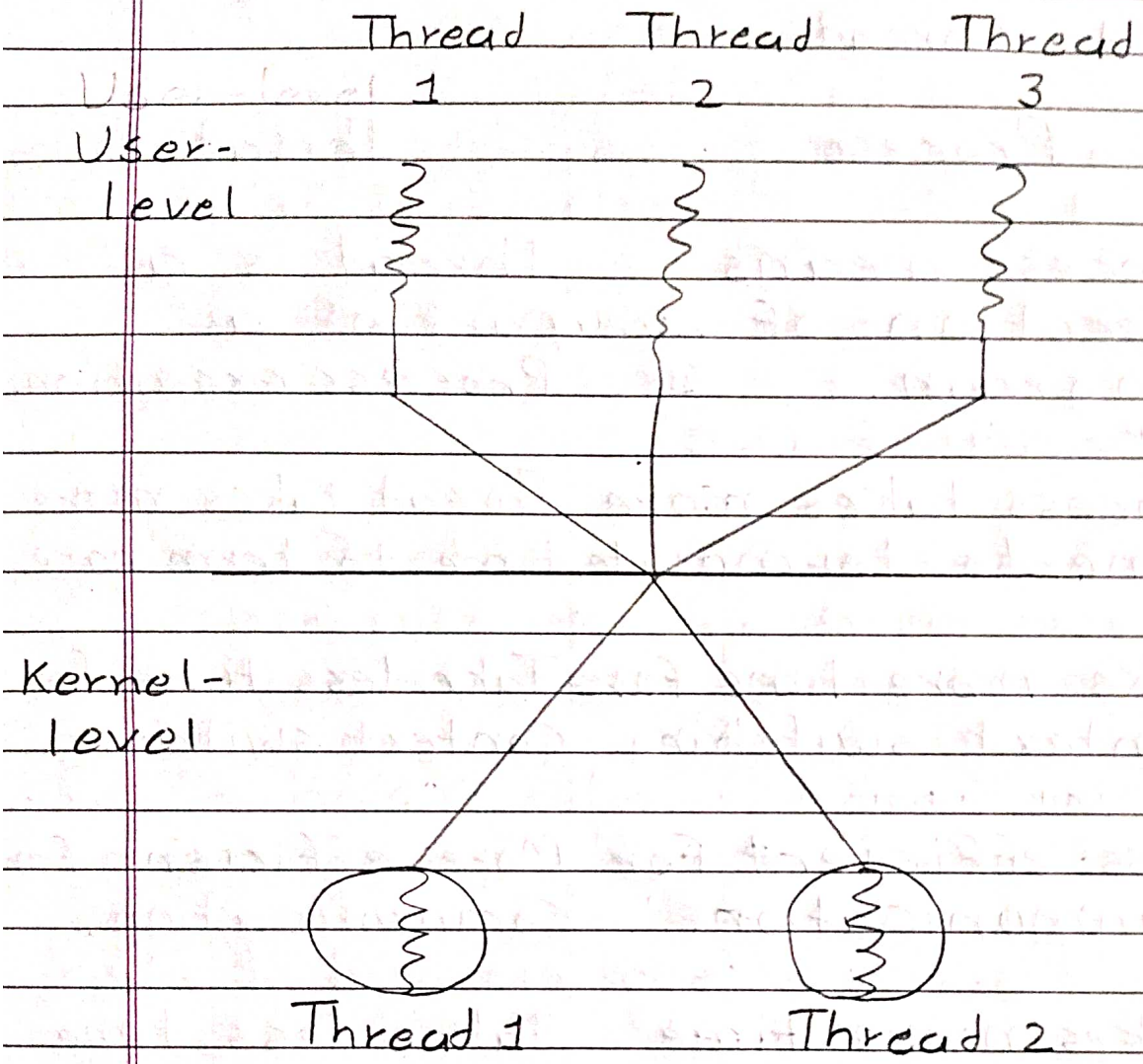
In this model, one user level thread maps to one kernel level thread



In this Diagram, One thread is connect with one kernel-level thread.

3 Many to Many Multithreading.

In this model, many user level thread maps to many kernel-level thread.



In this diagram, Many User-level thread is connect with same or much less variety of Kernel level thread.

* Difference Between Process and Thread:

	Process	Thread
1	Process means execution of Program.	Thread is a part of a Process execution.
2	Process takes more time to terminate.	Thread takes more time to terminate.
3	Take more time for Context switching.	Take less time for Context switching.
4	Less efficient For Communication.	More efficient for Communication.
5	Takes more time for creation.	Takes less time for creation.
6	Process is not share memory.	Thread is share memory.
7	Heavyweight	Lightweight
8	Process does not share data. each other	Thread share data with each other.

* Difference between Preemptive and Non-Preemptive algorithm.

	Preemptive	Non-Preemptive
1	CPU cycle are allocated to a process for some time.	CPU cycle are allocated to a process for require burst time.
2	Two Process can interrupt in between.	Two Process can not interrupt in between.
3	It has Overheads Scheduling.	It does not have Overheads scheduling
4	Less Waiting time	More Waiting time
5	CPU Utilization is high.	CPU Utilization is low.
6	Respones time is less	Respones time is high.
7	Cost associated	No cost associated
8	Ex. Round Robin, SRTN	FCFS, SJF

Page No.
Date : / /

* Explain First Come First Serve Algorithm.

=> First come First serve algorithm is a Non-Preemptive algorithm

This Process come First in a ready queue this process get CPU First.

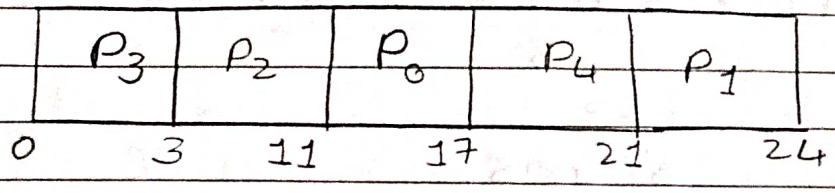
IF Process have only Burst time then which Process come in the First this Process get CPU First.

IF Process have Burst time and Arrival th time then which Process has less arrival time this Process get CPU First.

Ex.

Process	Burst time	Arrival time
P_0	6	2
P_1	3	5
P_2	8	1
P_3	3	0
P_4	4	4

Grantt Chart:



Here, Process P_3 have less arrival time. So, P_3 get CPU First.

After that according to Arrival time P_2, P_0, P_4 and P_1 get CPU resepectvly.

Process table:

Time	Process	Turnaround = P.C - P.A	Waiting = P.T - P.B
0	-	-	-
3	P_3	3	0
11	P_2	10	2
17	P_0	15	9
21	P_4	17	13
24	P_1	19	16

Average Turnaround Time $= \frac{3+10+15+17+19}{5}$

$= 12.8$ ms

Average Waiting Time $= \frac{2 + 9 + 13 + 16}{5}$

$= 8$ ms

Throughput $= \frac{5}{24} = 0.208$ ms

CPU Utilization $= \frac{5}{(5+0)} * 100$

$= 100\ \%$

* - Advantage :

* Simple to implement

- Disadvantage :

Throughput is not efficient.

* Explain Shorted Job First algorithm.

=> Shorted Job First algorithm is Non-Preemptive algorithm.

Shorted Job First is a algorithm in which small Burst time Process get First CPU.

If Process have only Burst time then, smallest Burst time process get First CPU.

If Process have Burst time and arrival time then, First Process selected with smallest arrival time after that secouns Process selected with smallest Burst time Process.

Ex.

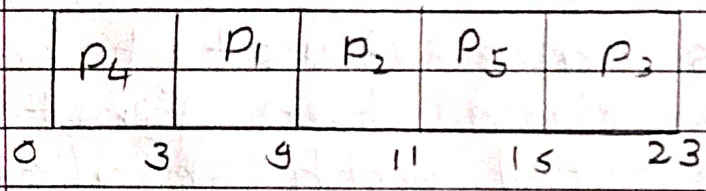
Process	Burst time	Arrival time
P_1	6	2
P_2	2	5
P_3	8	1
P_4	3	0
P_5	4	4

Here, Process P_4 have smallest arrival time. so, P_4 get CPU First.

After that Process have smallest Burst time, this Process get CPU First.

According to smallest Burst time P_2, P_1, P_5 and P_3 get CPU respectively.

Grantt Chart:



Process Table:

Time	Process	Turnaround Time = (P.C.T - P.A.T)	Waiting time = (P.T - P.B.T)
0	-	-	-
3	P_4	3	0
9	P_1	7	1
11	P_2	6	4
15	P_5	11	7
23	P_3	22	14

Average Turnaround Time $= \frac{3+7+6+11+22}{5}$

$= 9.8$ ms

Average Waiting Time $= \frac{0+1+4+7+14}{5}$

$= 5.2$ ms

Throughput $= \frac{5}{23} = 0.217$

CPU Utilization $= \frac{5}{(5+0)} * 100$

$= 100\%$

- Advantage: Less Waiting time
- Disadvantage: Difficult to estimate time required to complete execution.

* Explain Short Reamning time First algorithm.

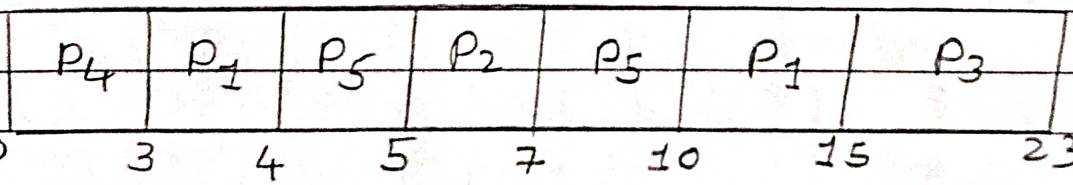
=> Short Reamning time First algorithm is a Preemptive algorithm.

In SRTN, Process put in the Queue according to its arrival time. after that which process have small burst time this process is execute first.

Ex.

Process	Burst Time	Arrival time
P_1	6	2
P_2	2	5
P_3	8	1
P_4	3	0
P_5	4	4

Gantt Chart:



Process table:

Time	Process	Turnaround T. (CP.C - P.A)	Waiting T. (CP.T - P.B)
0	-	-	-
15	P_1	13	7
7	P_2	2	0
23	P_3	22	14
3	P_4	3	0
10	P_5	6	2

Average Turnaround Time = $\frac{13+2+22+3+6}{5}$

= 9.2 ms

Average Waiting Time = $\frac{7+0+14+0+2}{5}$

= 4.6 ms

Throughput = $\frac{5}{23}$ = 0.217

CPU Utilization = $\frac{5}{5+0} * 100$

= 100 %.

- Advantage: Minimizes Waiting Time
- Disadvantage: SRTF has higher overload than SJF

* Explain Round Robin algorithm.

=> Round Robin is a preemptive algorithm.

If Process have only Burst time and Burst time and arrival time then those Process come First this Process get CPU First according to time quantum.

Ex.

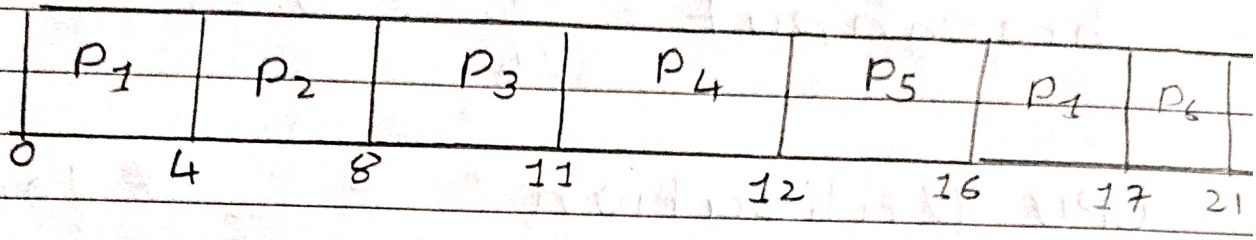
Process	Burst time	Arrival time
P_1	5	0
P_2	6	1
P3	3	2
P4	1	5
P5	5	6
P6	4	3

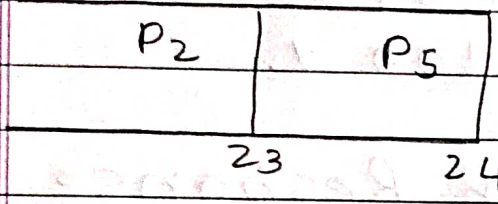
Time Quantum = 4

Page No.
Date : / /

Here, Process P_1 come First
So, Process P_1 get CPU First
For a limited time quantum.

Gantt Chart:





Process Table:

Time	Process	Turnaround (CP.C - P.A)	Waiting tim (CP.T - P.B)
0	-	-	-
17	P_1	17	12
23	P_2	22	16
11	P_3	9	6
12	P_4	9	8
24	P_5	20	15
21	P_6	15	11

Average Turnaround Time $= \frac{17 + 22 + 9 + 9 + 20 + 15}{6}$

$= 15.33$ ms

Average Waiting Time = $\frac{12+16+6+8+15+11}{6}$

= 11.33 ms

Throughput = $\frac{6}{24}$ = 0.25

CPU Utilization = $\frac{6}{(6+0)}$ * 100

= 100 %.

- Advantage : Faster Response Time

- Disadvantage : High average Waiting time.

* Explain Priority algorithm.

=> Priority algorithm is a non-preemptive algorithm.

If Process have only Priority then those Process have higher Priority this Process get CPU First.

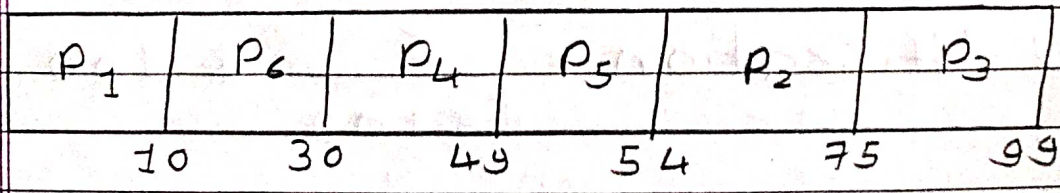
IF Process have Priority and arrival time then those Process have smallest arrival time time Process get CPU First.

IF Process have Same Priority and same arrival time and Burst time then those Process have smallest Burst time this Process get CPU First.

Ex.

Process	Burst Time	Arrival Time	Priority
P_1	10	3	2
P_2	21	7	6
P_3	24	1	9
P_4	19	3	3
P_5	5	8	4
P_6	20	8	2

Gantt Chart:



Process Table:

Time	Process	Turnaround	Waiting
0	-	-	-
10	P_1	10	0
30	P_6	30	10
49	P_4	49	30
54	P_5	54	49
75	P_2	75	54
99	P_3	99	75

Average Turnaround Time = $\frac{10+30+49+54+75+99}{6}$

= 52.83 ms

Average Waiting Time = $\frac{10+30+49+54+75}{6}$

= 36.33 ms

Throughput: 6 / 99 = 0.06 ms

CPU Utilization = $\frac{6}{(6+0)} * 100$

= 100 %.

- Advantages : Simple implementation.

- Disadvantages : Starvation is possible for low priority process.

3 Explain different State of a Process with diagram.

Process is called execution of a Program.

For execute any Process, there are use Process state diagram.

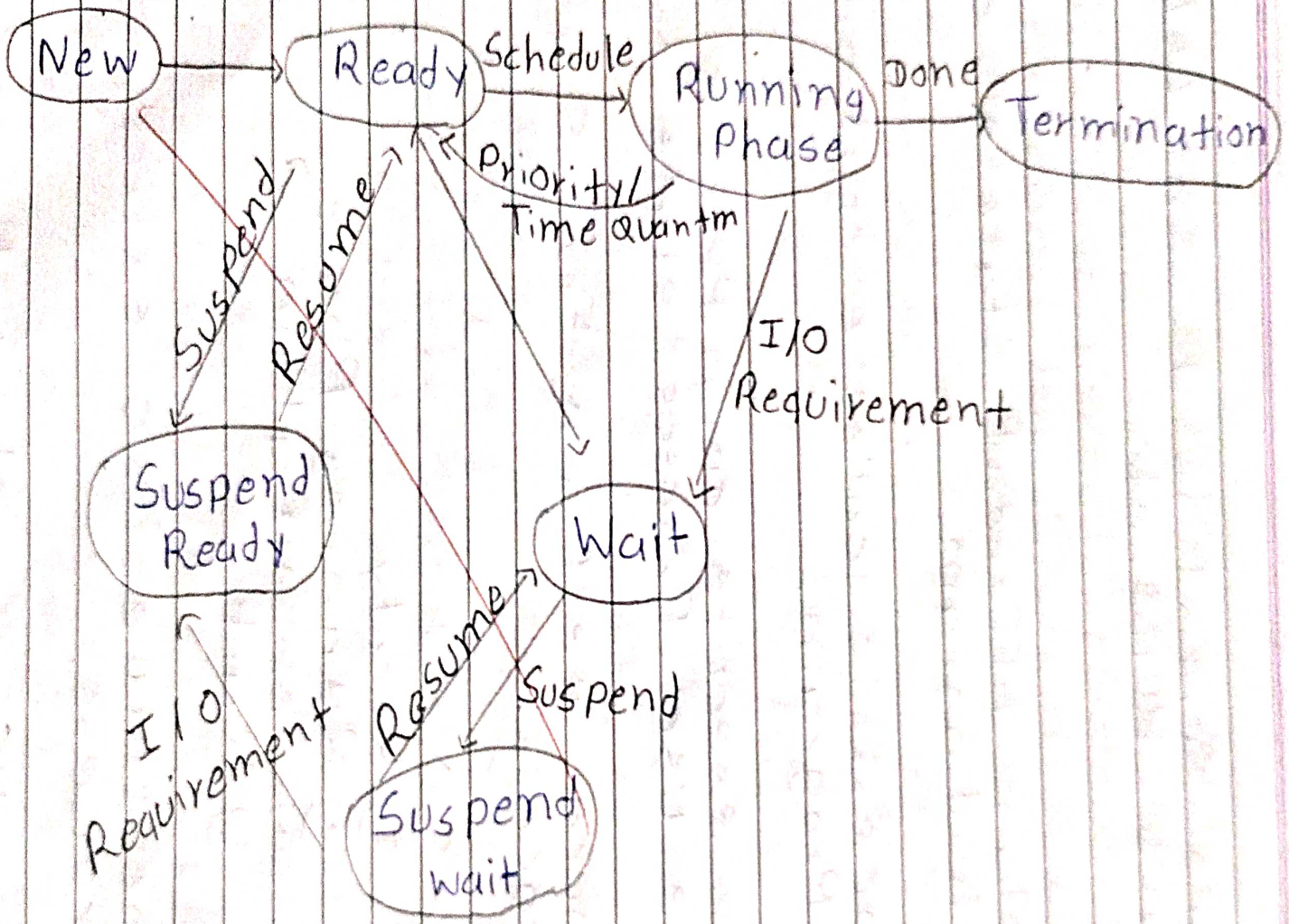
There are Main Five State of Process.

1) New
2) Ready
3) Running Phase
4) Termination
5) Wait

There are Two extra state in Process.

1) Suspend Wait
2) Suspend Ready.

Explain 7 state process model with neat diagram. (8 Marks)



SMVS
Page No.
Date : / /

1 New State: This is a First state of Process.

In new state, Process is created and go in the Ready state.

2 Ready State: This is Second state of Process.

In Ready state, Process execution are start and go in the Running Phase.

If Process Require more time than process go in the suspend Ready state and after that Process Resume and go in the Ready state.

3 Running Phase: This is Third state of Process.

In Running Phase, Process Final execution is done.

If Process have Priority or Time Quantnm than Process go in the Ready state.

If Process Requrie Some Input-output than Process go in the wait state.

If Process execution is done than Process is go in the Termination state.

4 Wait State : This is fourth state of Process.

If Process requrie more time than Process go in the Suspened state and after that Process Resume and go in the wait state and after that Process go in the Ready state.

5 Termination state:

After the Process execution Process pass in this state.

After this state Process is complete.

- Suspend Wait:

In waiting state, if Process require more time then Process pass in this state.

If In this state, Process have any I/O Requirement then Process go in the Suspend Ready state.

- Suspend Ready:

In Ready state, if Process have more time than Process go in the Suspend Ready state.

Page No.
Date : / /

4 Explain Life Cycle of Thread.

Instruction of Process is called Thread.

There are Five state of Thread Life Cycle.

1) Ready
2) Running
3) Wait
4) Sleep
5) Death / Termination.

1 Ready : This is First state of Thread.

Execution of Thread is start after the Ready state.

After this state, Thread go in the Running state.

Death/ Termination
Running
Done
execution
Done (Timed)
Requirement
Done Requirement
Sleep
Ready
I/O Require
Wait
Completion

If ~~Sta~~ Thread have Input output Requirement than thread go in the wait state.

If Thread have other requirement than thread go in the sleep state.

After the Running Phase execution of thread is complete than thread is go in the sleep state.

2 Running:

After the Ready Phase, execution of thread is done in this state.

3 Wait:

In this state, Input-output Requirement of Thread is Done.

SMVS
Page No.
Date : / /

4 Sleep:

If thread have any other requirement than thread is go in the this state.

5 Death:

If thread execute Perfectly then thread go in the state.