Inter Process Communication (IPC)

Inter process communication is a mechanism for processes to communicate and to synchronize their actions.

There are two main models of IPC

1. Shared memory: A region of memory is shared between processes. Each can read/write to it.

2. Message passing: Processes send and receive messages to communicate without shared memory.

If processes don't share memory, they use message passing (pipes, queues, sockets). If they can share memory, they use shared memory and synchronize mechanisms. OS provides IPC facilities to make this safe and efficient.

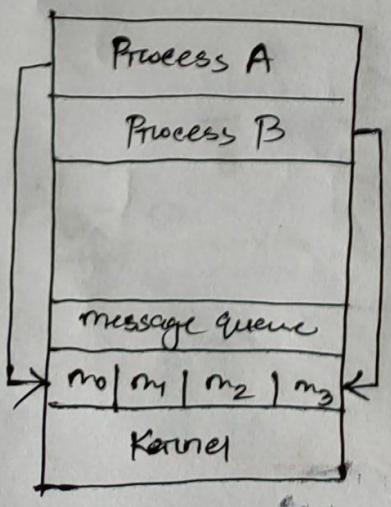


Fig: Message passing

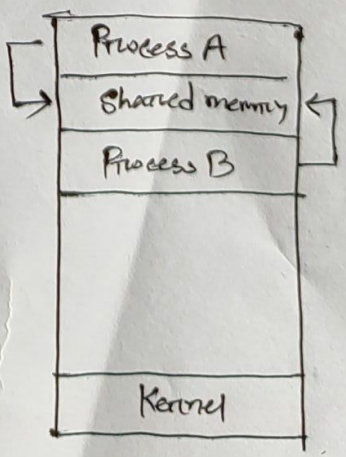


Fig: Shared memory

Remote Procedure Call (RPC)

Remote Procedure Call (RPC) is a methode that allows a program on one machine (client) to call a procedure on another machine (server) .like a local procedure call.

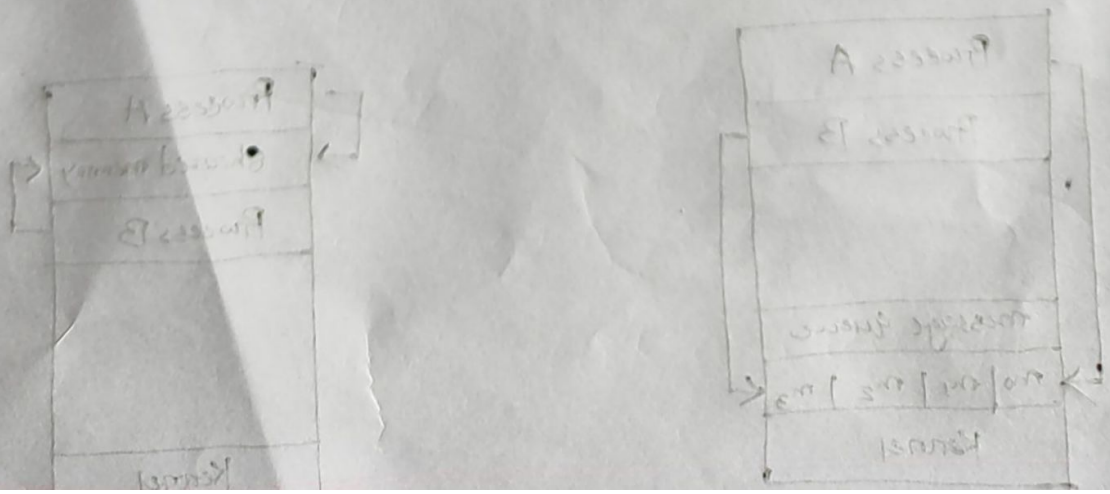
It hides the details of network communication using stubs and message passing to transfer data between client and server.

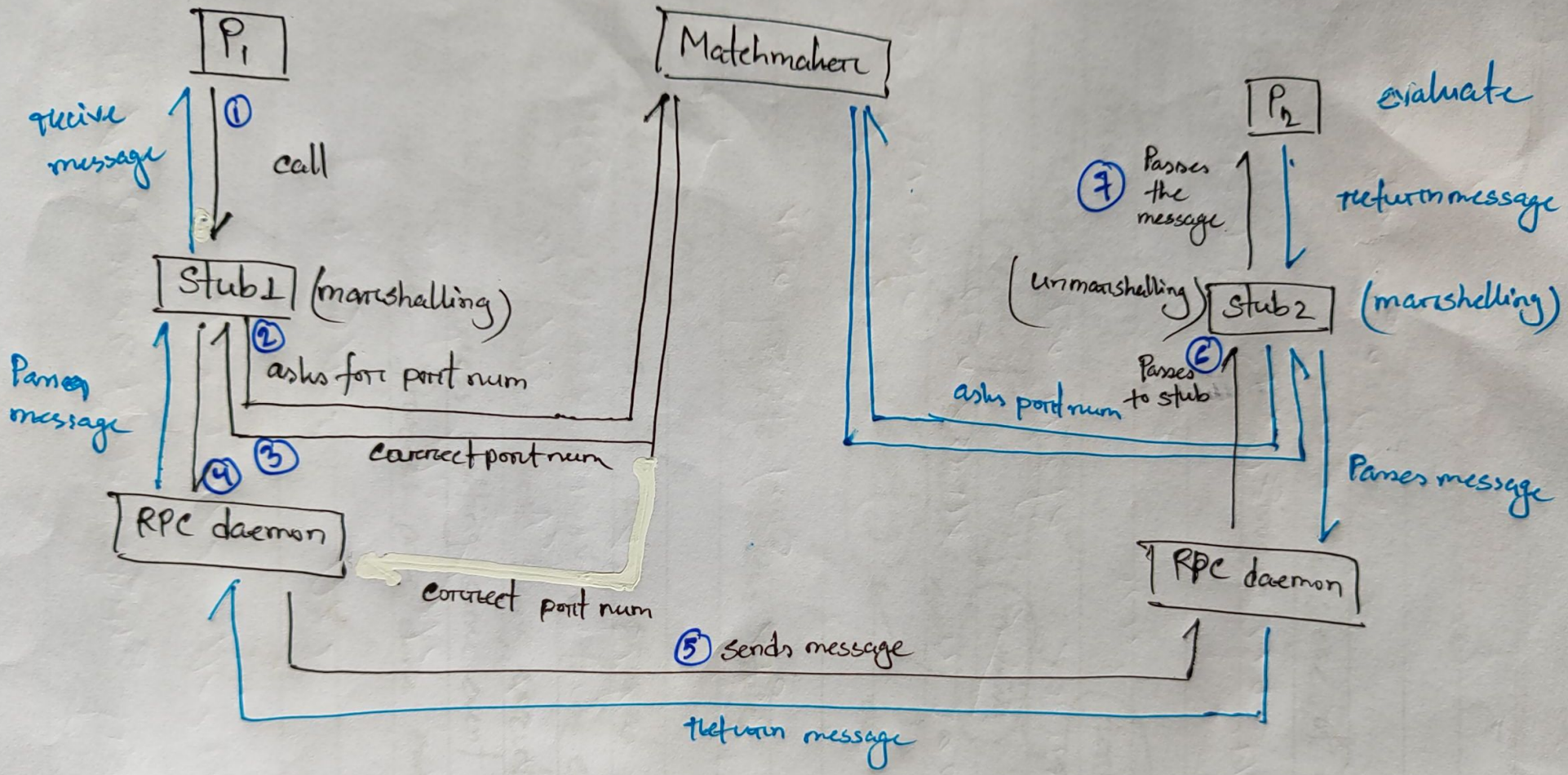
The remote procedure call contains

Stub - acts as local proxy, Pacls & unpacls (marshalls and unmarshalls) messages

Matchmaker - Finds correct server and port

Deamon - Acts as listener component




P1
Matchmaker
P2
evaluate
receive message
1
call
7
Passes the message
return message
Stub1
(marshalling)
(unmarshalling)
Stub2
(marshalling)
2
asks for port num
6
Passes to stub
asks port num
Passes message
3
correct port num
4
RPC daemon
RPC daemon
correct port num
5
sends message
return message

বিষয় : .....

তারিখ : ..... সময় : .....

Process State

What is Process State?

Process state represents the stage of execution of a process at any given time.

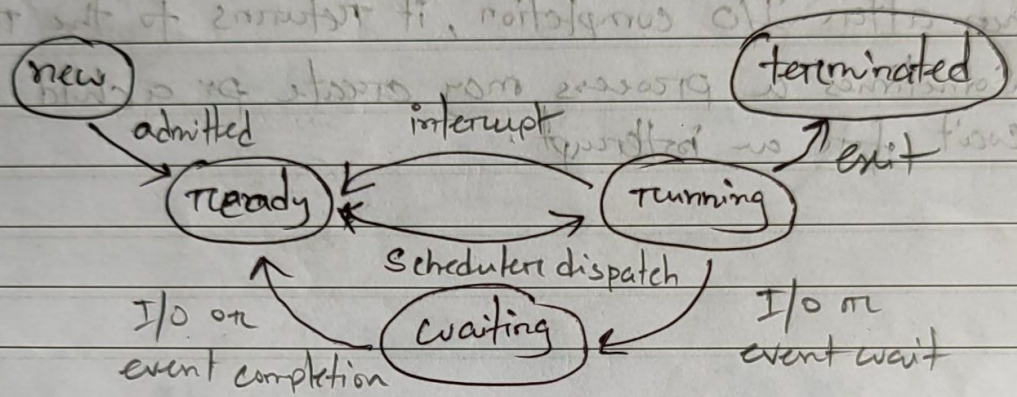


Figure: Process State

How its working?

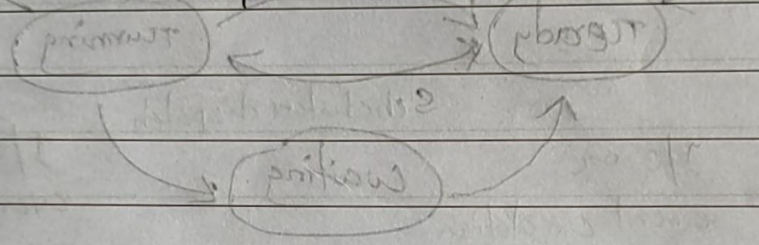
1. New - Process is being created
2. Ready - Process is waiting to be executed by the CPU
3. Running - Process are being executed by the CPU
4. Waiting - Process waiting some event to finish.
5. Terminated - Process are done ~~comp~~ execution.

বিষয় :

তারিখ : ..... সময় : .....

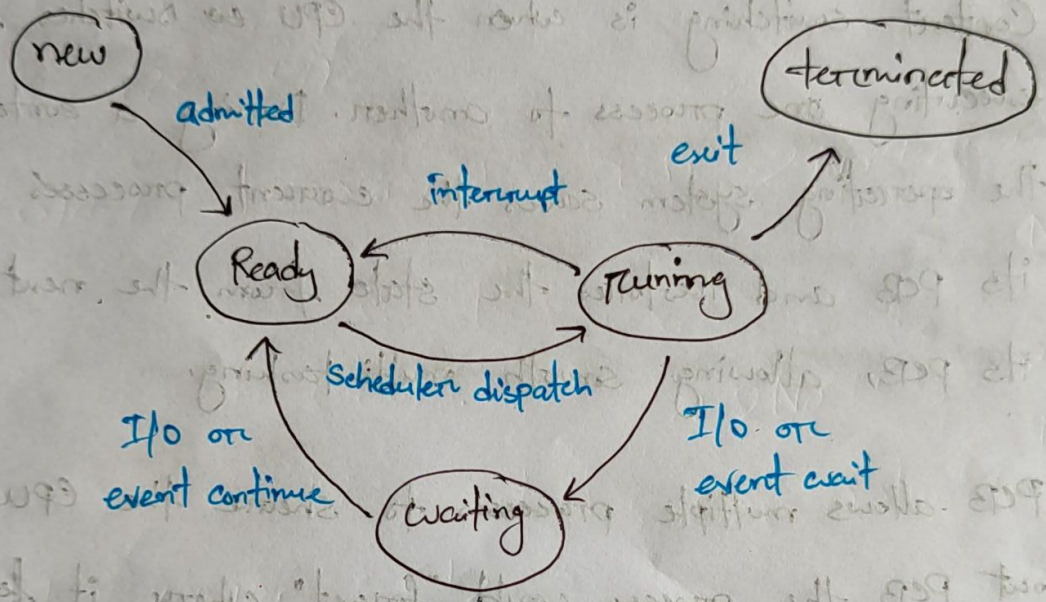
How it works?

1. Processes first go into ready queue.
2. The cpu executes one process at a time.
3. If the time slice expires → process goes back to Ready Queue.
4. If the process needs I/O → it moves to I/O queue, waits then after I/O completion, it returns to the ready queue.
5. Sometimes a process may create a child process or wait for an interrupt.



Process State

Process state represents the stage of execution of a process at any given time.



As a process executes, it changes state.

New: Process is being created

~~**Running:**~~

Ready: Process is waiting to be executed by the CPU

Running: Process instructions are being executed by the CPU

Waiting: Process waiting for some event to finish.

Terminated: Process are done execution

Process Control Block (PCB)

—A process control Block or PCB is a data structure used by the operating system to manage process by storing critical information about each processes execution

Context switching is when the CPU switches from executing one process to another. During a context switch, the operating system saves the current processes state in its PCB and restores the state from the next process from its PCB, allowing smooth multi-tasking.

The PCB allows multiple processes to share the CPU efficiently. Without PCB, the process would "forget" where it left off. Since each process has its own PCB, they don't interfere with each other.

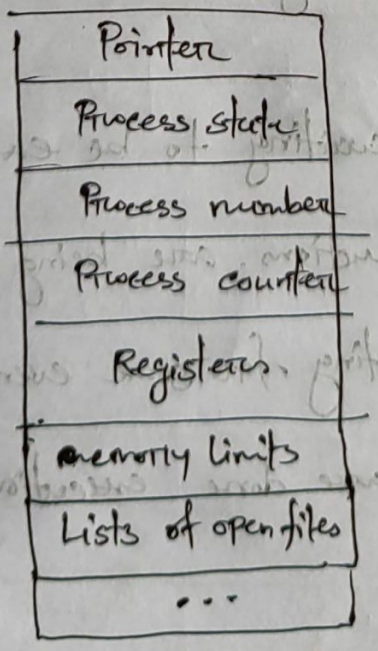


Fig: Process Control Block (PCB)

Context switch

A context switch a is a mechanism which allows the CPU to switch execution between processes, ensuring multitasking and efficient CPU utilization.

Steps:

1. Save the state of the current running process
2. Load the save state of the next scheduled process.
3. Continue execution of the new process from where it left off.

Socket.

A socket is an endpoint for communication between two machines over a network.

Socket = IP Address + Port

How it works?

Client creats a socket

↓

Server then a socket open

↓

Connection happens

↓

Data is sent and received

↓

connection closes

Process Scheduling

Process schedulin is is mechanism of deciding which process will run on the CPU next.

It chooses processes from ready queue.

- Types of Scheduling Queue:

1. Job queue: set of All processes in the system.
   - ~~sometimes the only sche~~
2. Ready queue: set of processes in the memory waiting to be executed.
3. Device queues set of processes waiting for an I/O device

Representation of process scheduling

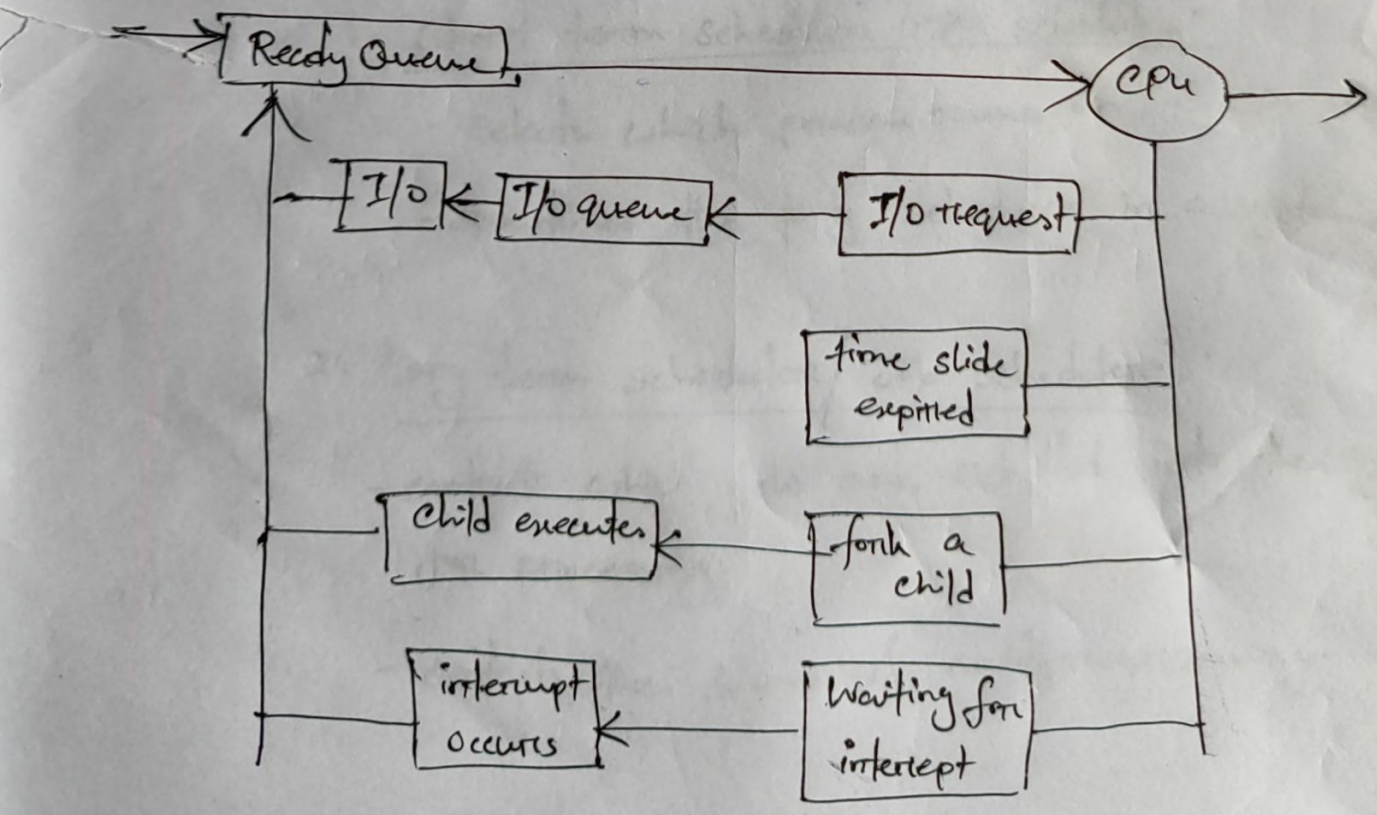


Fig: Queueing diagram.

Scheduler

A scheduler is a key component of an operating system that determines which process should run at any given time. The main role is to optimize CPU utilization.

Scheduler decides which process runs on the cpu, which process goes to the ready queue.
which waits for I/O

Schedulers are critical in time-sharing system and multiprogramming environments.

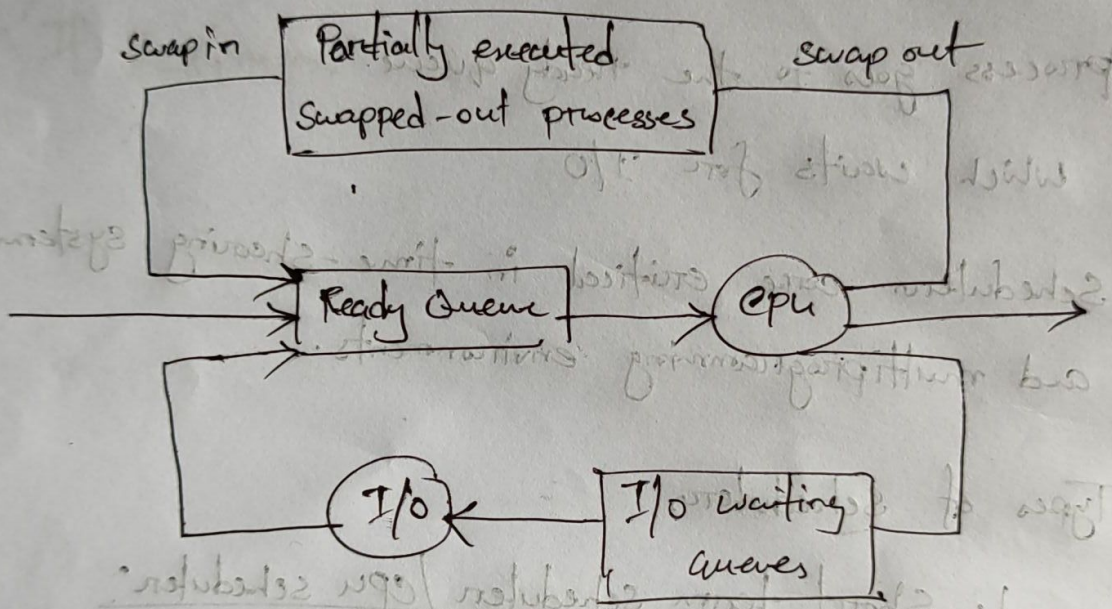
Types of schedulers:

1. Short term scheduler / CPU scheduler:
   - selects which process runs on the cpu next.
   - sometimes the only scheduler in a system.

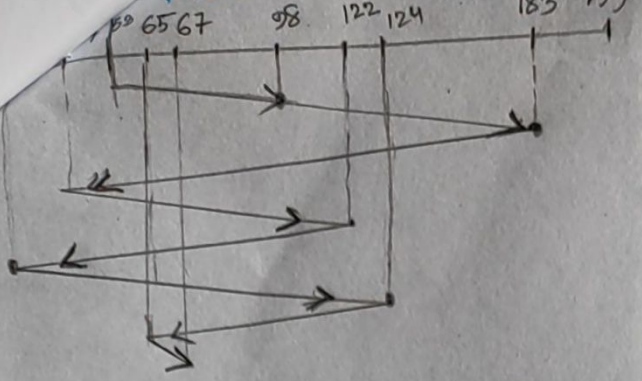
2. Long term scheduler / Job scheduler:
   - controls which jobs are admitted into the system for processing.
   - controls the degree of multiprogramming.

iii) Medium term scheduler.

1. Suspends and resumes processes to balance CPU and I/O usage.

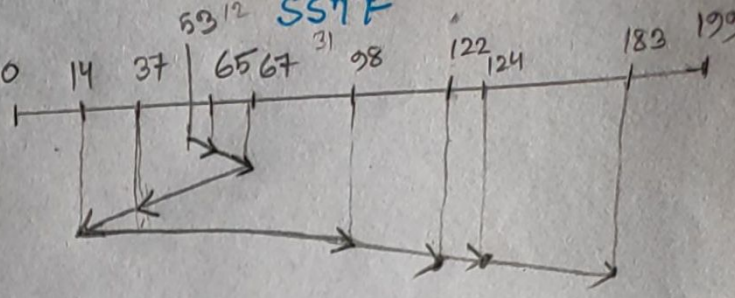


FCFS



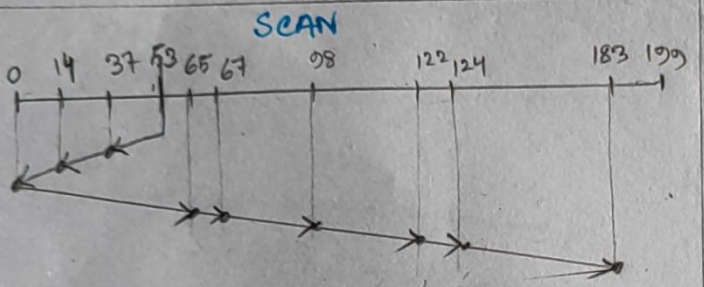
* যে আগে আসবে তাকে আগে Touch করবে

SSTF



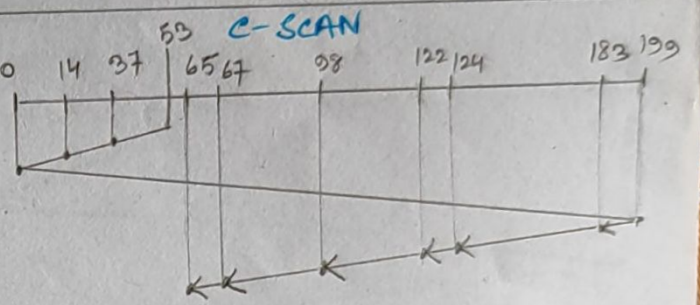
* কাছাকাছি যে আছে, তাকে আগে Touch করবে।

SCAN

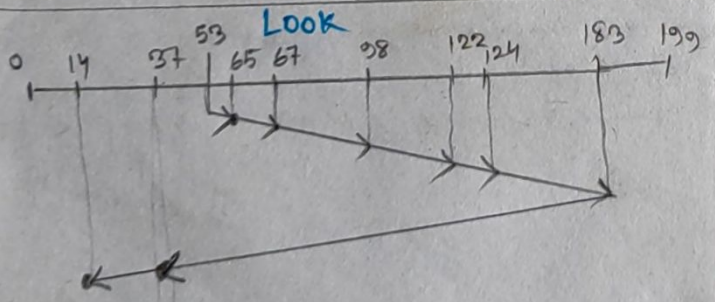


* যেদিকে যাবে, সবগুলো Touch করে edge এ যাবে, then আসার সে আকারে তাকে see-touch করবে.

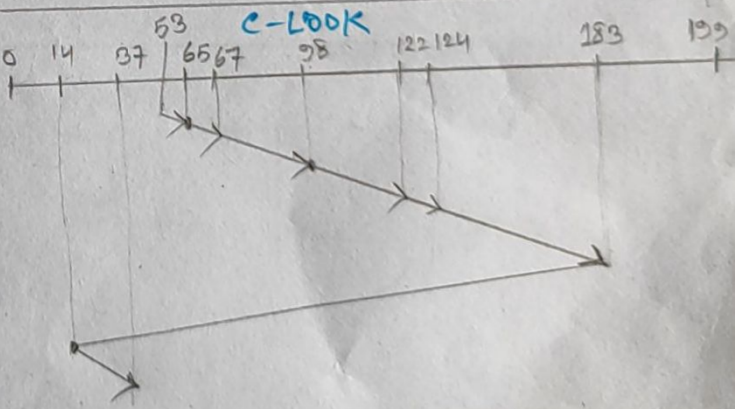
C-SCAN



LOOK



C-LOOK



Illustrate scheduling algo with a request queue

98, 183, 37, 122, 14, 124, 65, 67

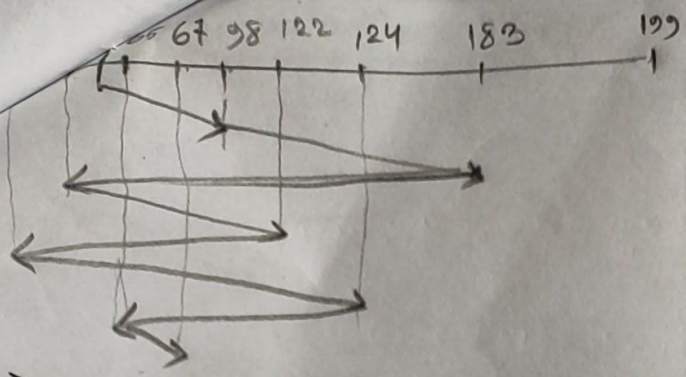
Head pointer 53

→ 14, 37, 65, 67 98, 122, 124, 183

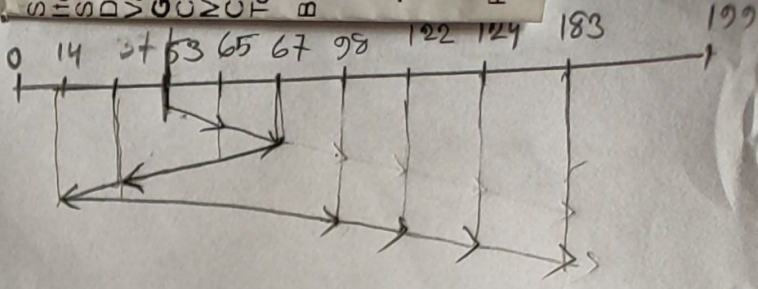
* Total num of head movement
=

* Total Time = Total num of head movement X per head movement

FCFS

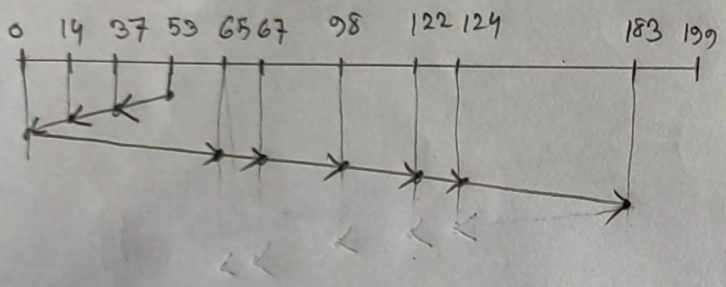


* যেটা আগে আসবে সেটায় আগে touch করবে।



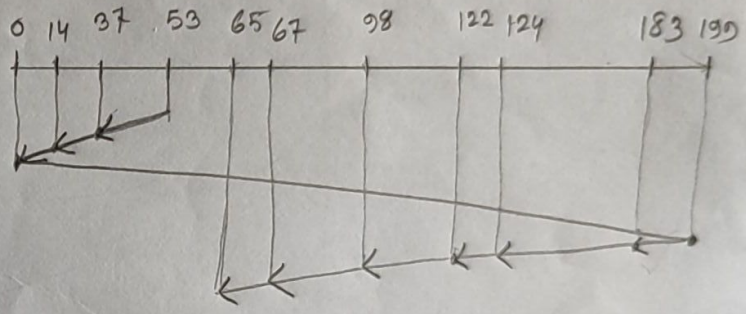
* starting point থেকে যেটা কাছে, সেখানে আগে যাবে।

SCAN



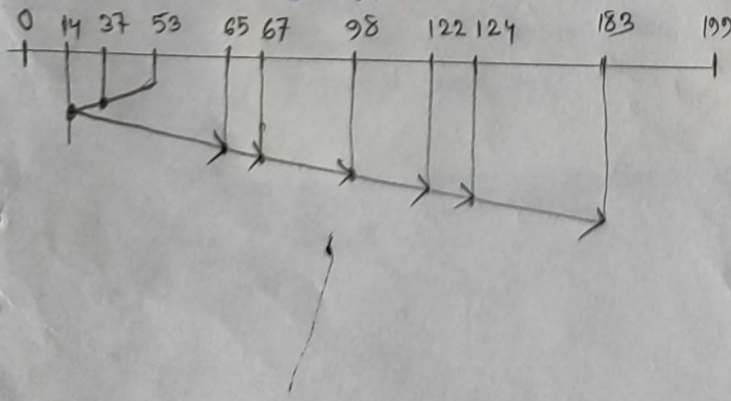
* যেদিকে প্রথম যাওয়া সুরু করবে, একদম শেষ মাথায় যাবে।

C - SCAN

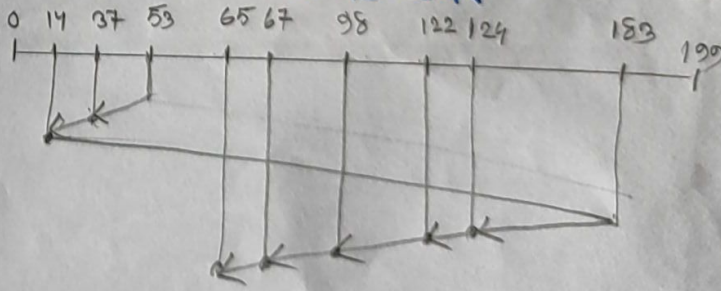


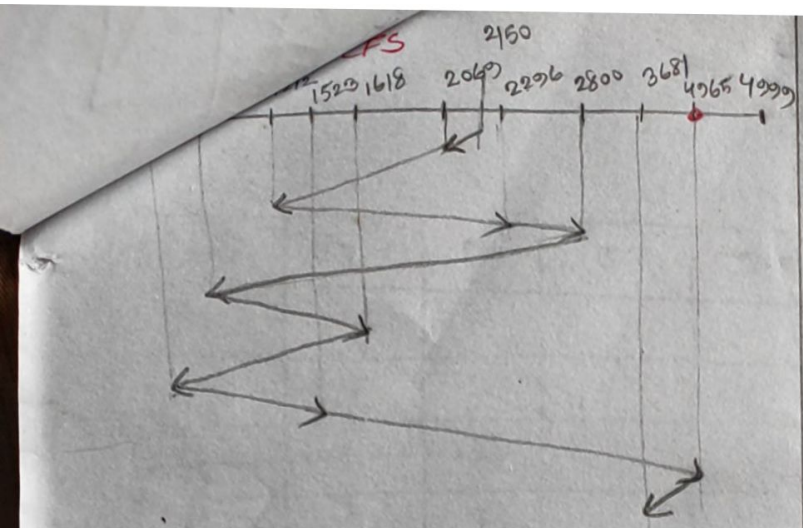
* দুই মাথার শেষ প্রান্ত touch করবে

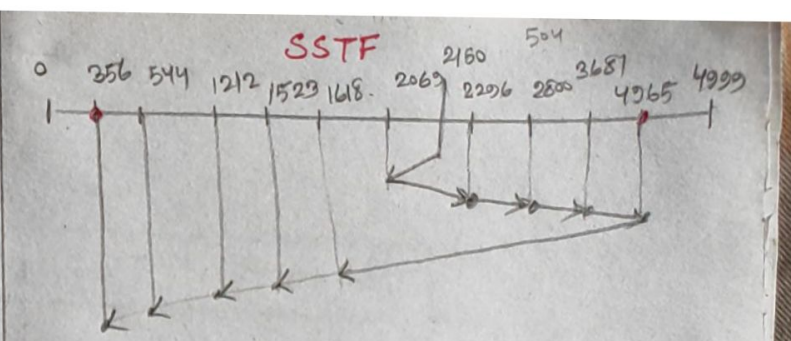
LOOK

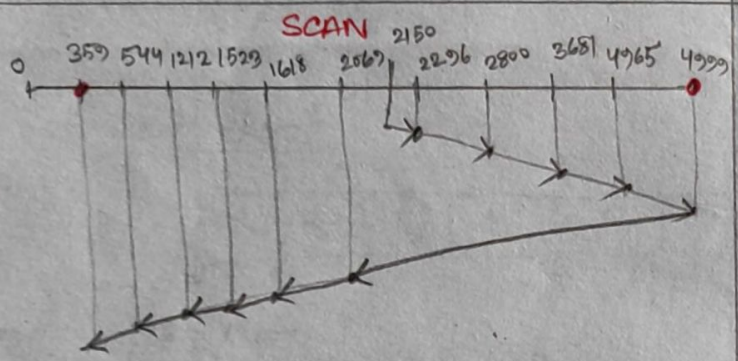


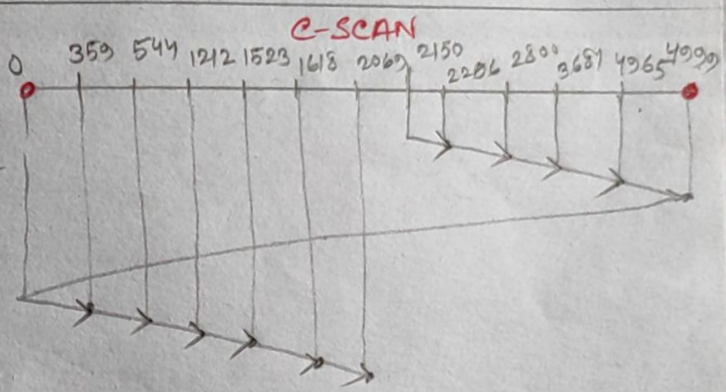
C-LOOK

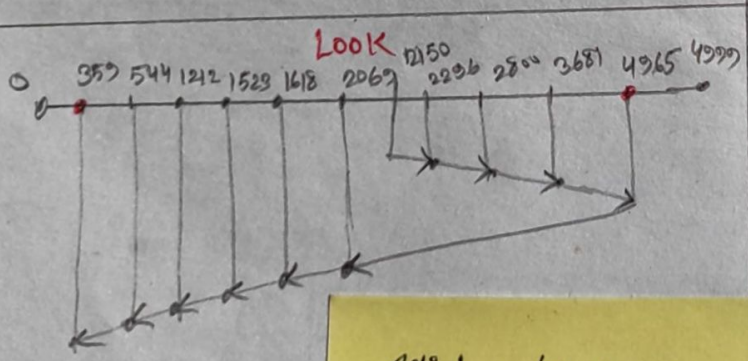


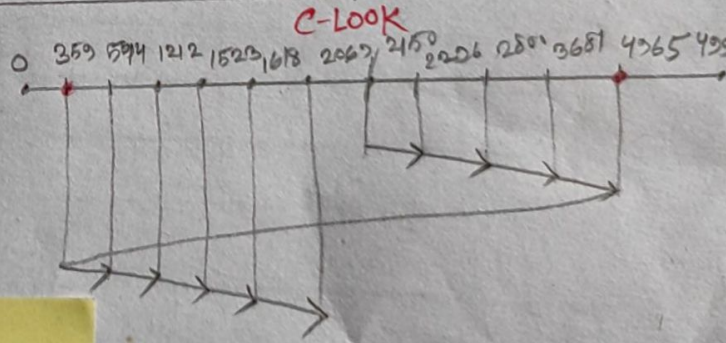












Cylinders (0-4999)

Head pointer 2150

2069, 1212, 2296, 2800, 544, 1618, 356, 1523, 4965, 3681

Sorted

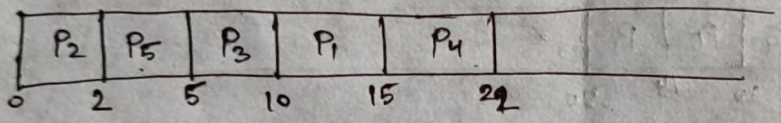
356, 544, 1212, 1523, 1618, 2069, 2296, 2800, 3681, 4965

SJF

(Non-Preemptive)

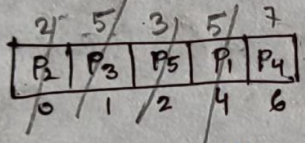
Process	AT	BT	CT	TAT	WaiT	ResponseT
P_1	4	5				
P_2	0	2				
P_3	1	5				
P_4	6	7				
P_5	2	3				

Gantt chart



* Burst time calculation অনুশীলন

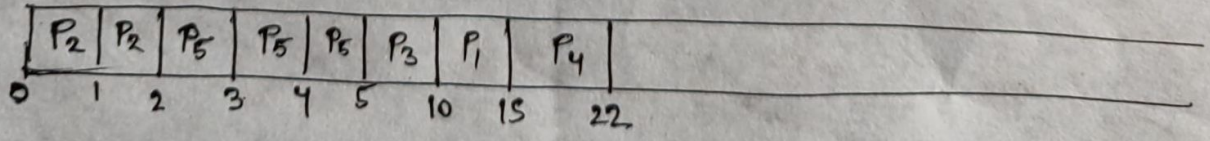
Ready Queue:



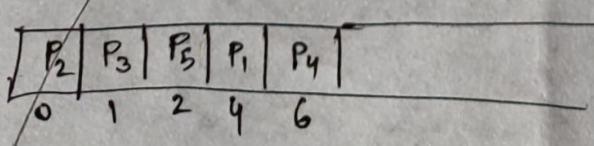
(Pre-emptive)

Process	AT	BT	CT	TAT	WAT	RT
P_1	4	5 0				
P_2	0	2 1 0				
P_3	1	5 0				
P_4	6	7				
P_5	2	3 2 1 0				

Gantt chart:



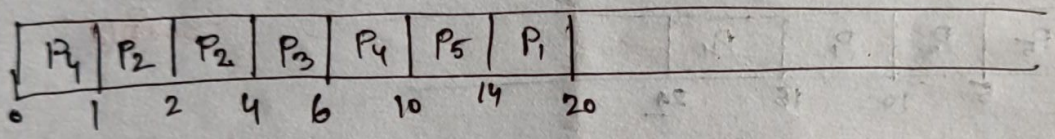
Ready Queue:



(SRTF)
(Pre-emptive)

Process	AT	BT	CT	TAT	WT	RT
P_1	2	6				
P_2	1	~~3~~ 2 0				
P_3	4	2 0				
P_4	0	~~5~~ 4 0				
P_5	6	4 0				

Grantt chard:



Operating System

Chapter-6

CPU Scheduling

CPU Scheduling: it is the process used by the operating system to choose which process/task will use the CPU next from the queue.

- It is the basis for multiprogrammed operating system. Where multiple processors are ready to run but only one can use the CPU at a time.
- The goal is to maximize the CPU utilization and ensure fair and efficient process management.

Terms we'll use:

Throughput – Number of processes that complete their execution per time unit. $\frac{\text{No. of Processes}}{\text{Total unit of Time}}$

Turnaround Time – amount of time to execute a particular process.

Completion time – Arrival time

Waiting Time – amount of time a process has been waiting in the ready queue.

Turnaround time – Burst time

Response Time – time from when a process arrives to when it starts executing.

Start of process execution – Arrival of process

Scheduling Algorithm Optimization Criteria

- Max CPU utilization
- Max throughput
- Min turnaround time
- Min waiting response
- Min response time

Turn Around Time = Completion - Arrival

Waiting Time = T.A.T - Burst Time

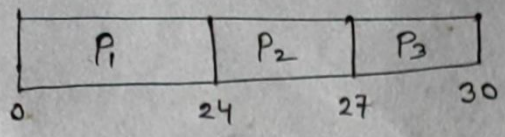
Response time = 1st time Processor gets the CPU - Arrival time

First Come First Served [FCFS]

Process ID	Burst Time	Arrival Time
P_1	24	0
P_2	3	0
P_3	3	0

Q: The order of arrival is P_1, P_2, P_3. Calculate Average Waiting time, Turnaround time and Avg Response time

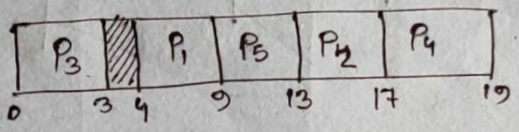
Gantt chart



ProcessID	Burst Time	Arrival Time	Turnaround Time	Waiting time	Response time
P_1	24	0	24-0=24	0	0
P_2	3	0	27-0=27	27-3=24	24
P_3	3	0	30-0=30	30-3=27	27
		~~Avg 27~~	Avg=27	Avg=17	Avg=17

Q.2.

Gantt chart



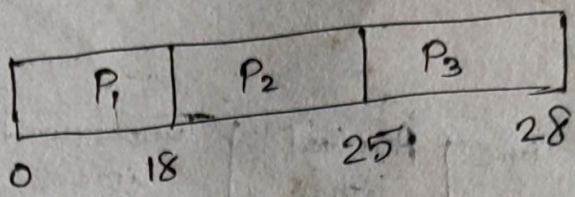
ProcessID	Burst Time	Arrival Time	Turnaround Time	Waiting time	Response Time
P_1	5	4	9-4=5	5-5=0	4-4=0
P_2	4	6	17-6=11	11-4=7	13-6=7
P_3	3	0	3-0=3	3-3=0	0-0=0
P_4	2	6	19-6=13	13-2=11	17-6=11
P_5	4	5	13-5=8	8-4=4	9-5=4
			Avg=8	Avg=4.4	Avg=4.4

ProcessID	Burst Time	Arrival Time
P_1	5	4
P_2	4	6
P_3	3	0
P_4	2	6
P_5	4	5

Q. Draw a Gantt chart of the Process. Calculate Avg waiting time, Turnaround time, Response time.

FCFS → always non-pre-emptive

Gantt Chart:



Process	BT Burst	AT Arrival	CT Completion time	TAT Turn Around Time	WT Waiting	RT Response
P_1	18	0	18			
P_2	7	0	25			
P_3	3	0	28			

$TAT = CT - AT$

$WT = TAT - BT$

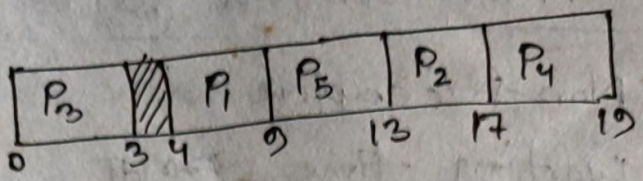
$RT = WT$ ← [Non-Pre-emptive]

$RT = ICT - AT$ ← Arrival Time [Pre-emptive]

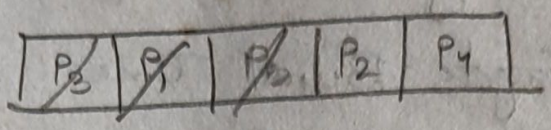
↳ Initial CPU Time

Practice

Gantt Chart



Ready Queue



Process	AT	BT	CT	TAT	WT	RT
P1	4	5	9	5	0	0
P2	6	4	17	11	7	7
P3	0	3	3	3	0	0
P4	6	2	19	13	11	11
P5	5	4	13	8	4	4
				Avg		

$$\text{Throughput} = \frac{5}{19} = 0.26$$

CPU

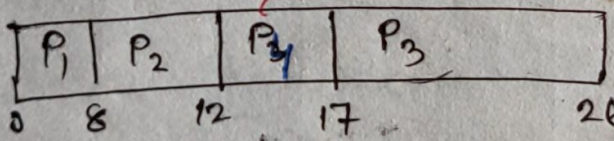
Shortest Job First (STF)

- Non-Preemptive
- waiting time কমানো
- Targets Burst Time

SJF (Slide Example)

Process	Arrival Time	Burst Time	Completion Time	TAT	WT	RT
P_1	0	8	8	8	0	0
P_2	1	4	12	11	7	7
P_3	2	9	26	24	15	15
P_4	3	5	17	14	9	9
				Avg: 14.25	Avg 7.75	Avg

Gantt Chart:



Throughput = $\frac{4}{26}$ = $\frac{\text{No of Process}}{\text{Total unit of time}}$

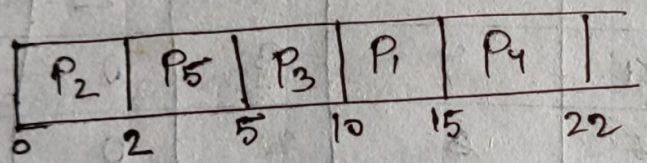
Ready Queue:

$= \frac{4}{26}$

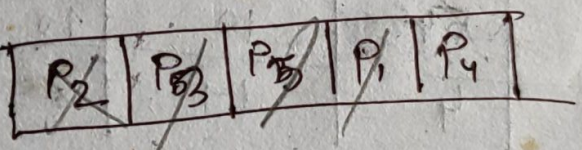
SJF Practice

Process	Arrival	Burst	Completion	TAT	WT	RT
P_1	4	5	15	11	6	6
P_2	0	2	2	2	0	0
P_3	1	5	10	9	4	4
P_4	6	7	22	16	9	9
P_5	2	3	5	3	0	0
				Avg = 8.2	Avg = 3.8	

Gantt chart:



Ready Queue:



~~From~~ Throughput = $\frac{5}{22}$

CPU overhead

SRTF : Shortest Remaining Time First

↳ SJF এর চালাক ভাই

SRTF	SJF
Pre-emptive	Non-pre-emptive

[1 TU]

Process	Arrival	Burst	Completion	TAT	WT	RT
P_1	2	6	20	18	12	14 − 2 = 12
✓ P_2	1	~~3 2 1~~	4	3	0	1 − 1 = 0
✓ P_3	4	~~2 1~~	6	2	0	4 − 4 = 0
✓ P_4	0	~~5 4 3 2 1~~	10	10	5	0 − 0 = 0
P_5	6	4	14	8	4	10 − 6 = 4

Gantt chart:

P_4	P_2	P_2	P_2	P_3	P_3	P_4	P_4	P_4	P_4	P_5	P_5	P_5	P_5	P_1	

0 1 2 3 4 5 6 7 8 9 10 14 20

0 থেকে শুরু হলে exception

Ready Queue: ~~P_4~~ ~~P_2~~ ~~P_1~~

P_4	P_2	P_1	P_3