CS & IT ENGINEERING

Operating Systems

1500 Series

Lecture No.- 01

By- Dr. Khaleel Khan sir

Recap of Previous Lecture

Topic

Topics to be Covered

- Topic
- Topic
- Topic
- Topic
- Topic

CPU-scheduling

[MSQ]

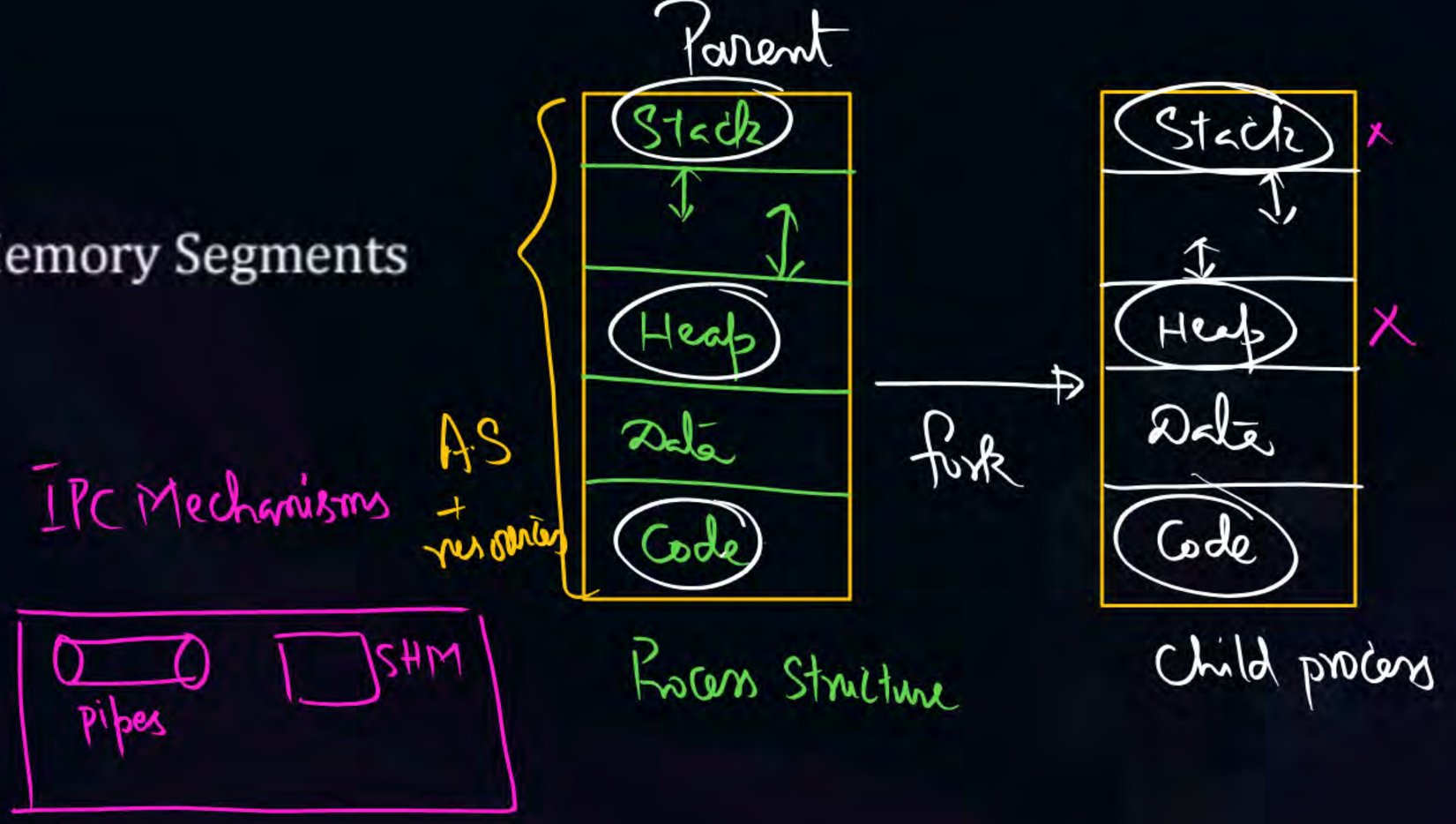
#Q. When a process creates a new process using the fork() operation, which of the following states is shared between the parent process and the child process?

A. Stack ×

B. Heap ×

C. Shared Memory Segments ✓

D. Pipes ✓

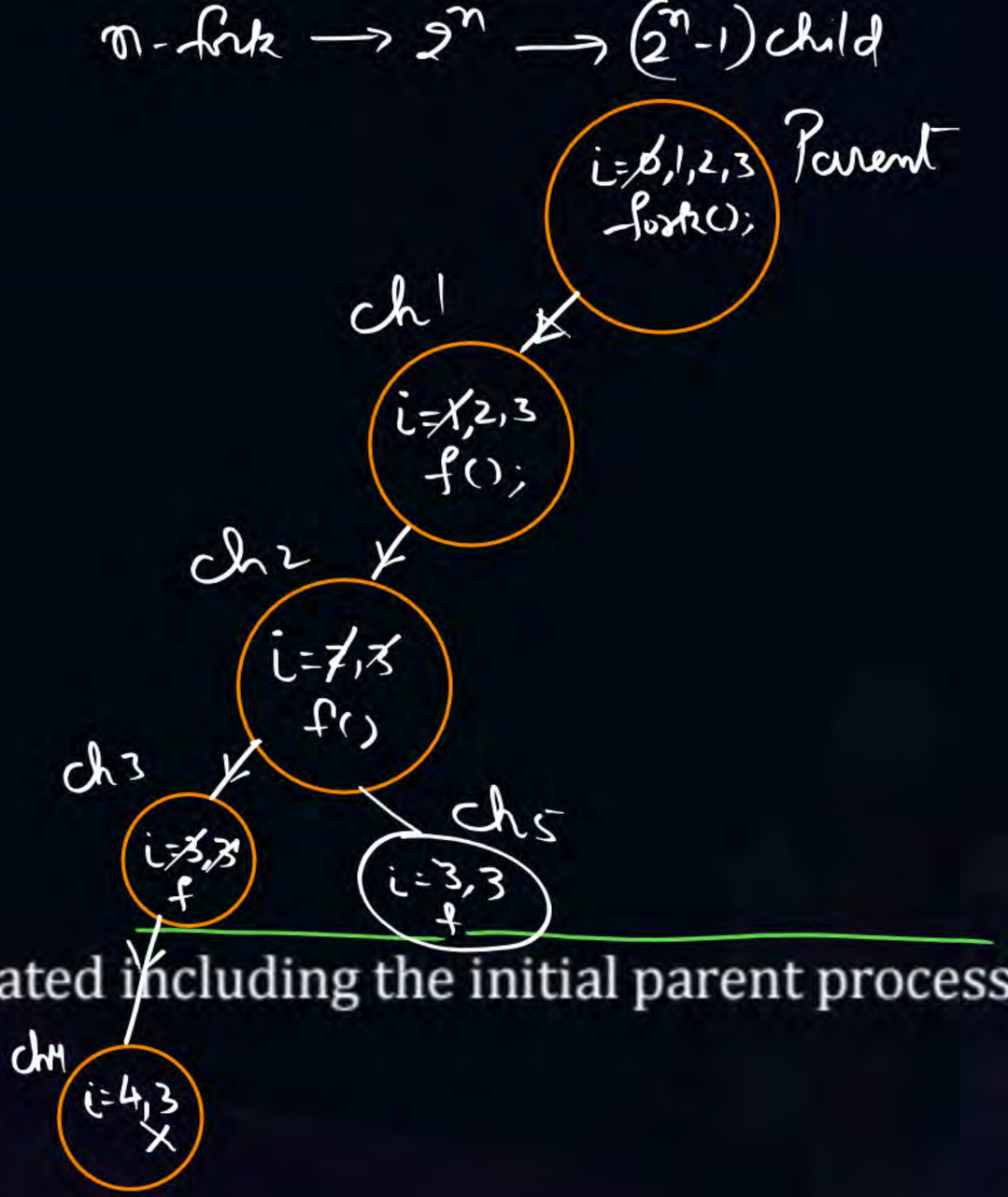


[NAT]

#Q.

```
#include <stdio.h>
#include <unistd.h>
int main()
{
        int i;
        for (i = 0; i < 4; i++)
                fork();
        return 0;
}
```

How many processes are created including the initial parent process?



[MSQ]

#Q. Which of the following components of program state are NOT shared across Threads in a Multithreaded process?

A Register values

B Heap memory

C Global variables

D Stack memory

Shared

Shared

<A ; D>

Thread

Stack + Register

[MCQ]

#Q. What system calls have to be executed by a command interpreter or shell in order to start a new process, which should execute an application?

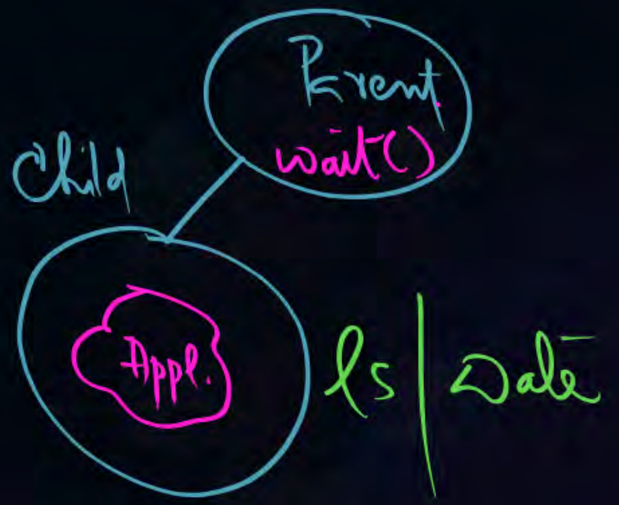
a) fork

b) fork, exec, getpid

c) fork, exec, wait

d) fork, exec, wait, exit

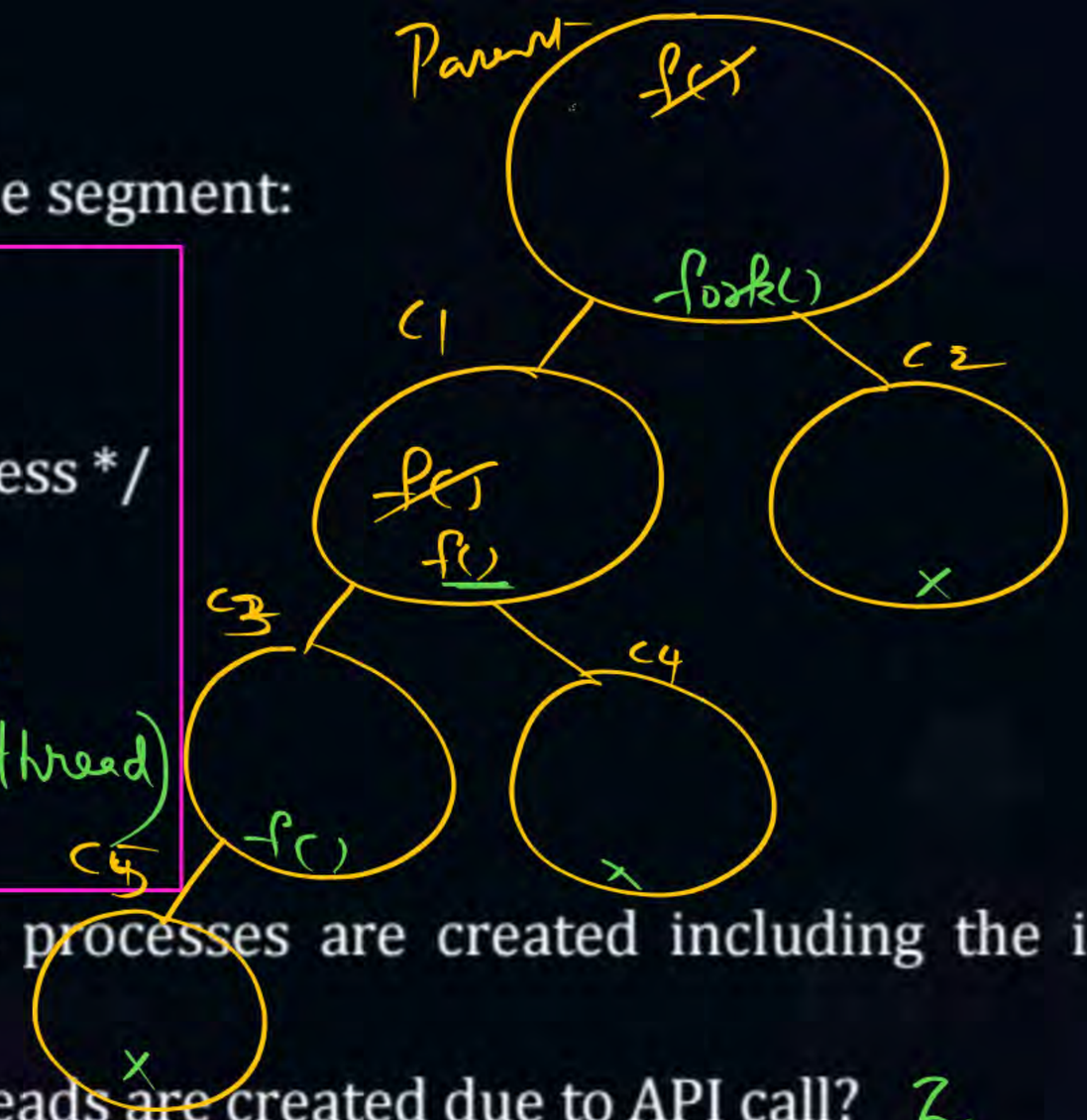
exec : is a System call to Execute a Command/ablf. → run a new Program



[NAT]

#Q. Consider the following code segment:

```
pid_t pid;
pid = fork();
if (pid == 0) { /* child process */
        fork();
        thread_create( . . . );
}
fork();
```



(a) How many unique processes are created including the initial parent process? 6

(b) How many new threads are created due to API call? 2

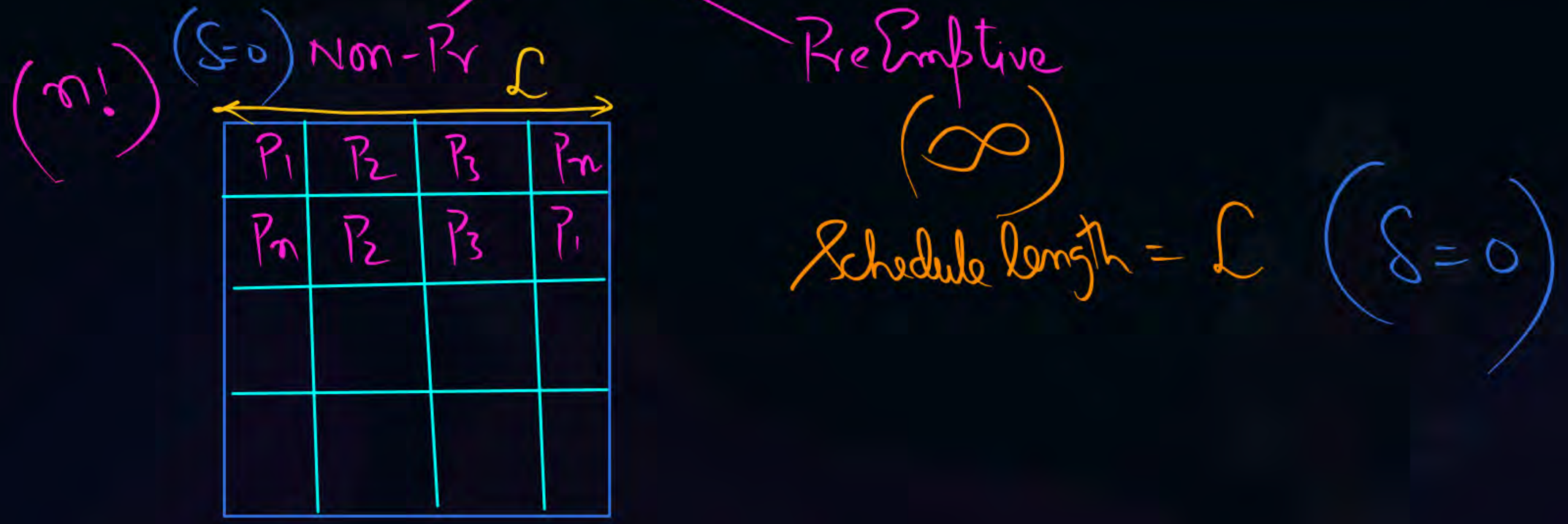
[NAT]

#Q. Which of the following instructions should be privileged?

a. Set value of timer.
b. Read the clock.
c. Clear memory.
d. Issue a trap instruction.
e. Turn off interrupts.
f. Modify entries in device-status table.
g. Switch from user to kernel mode.
h. Access I/O device

[NAT]

#Q. A CPU-scheduling algorithm determines an order for the execution of its scheduled processes. Given n processes to be scheduled on one processor, how many different schedules are possible? Give a formula in terms of n.



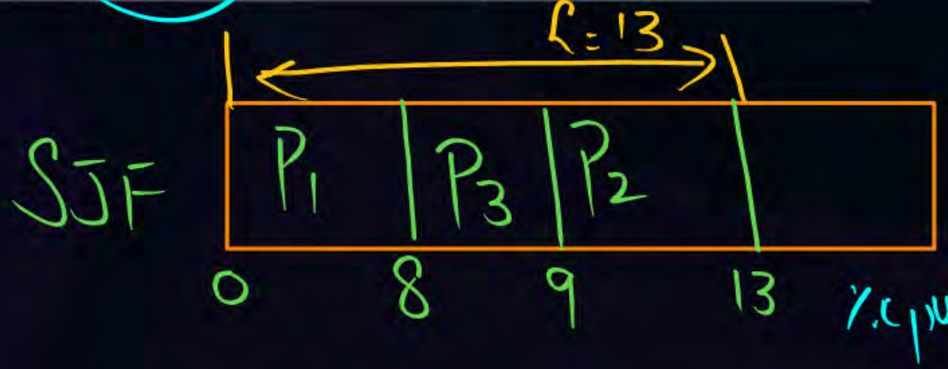
[NAT]

#Q. Suppose that the following processes arrive for execution at the times indicated. Each process will run for the amount of time listed. In answering the questions, use non-preemptive scheduling, and base all decisions on the information you have at the time the decision must be made.

Process	Arrival Time	Burst Time
P1	0.0	8
P2	0.4	4
P3	1.0	1

FCFS:

Av. TAT = 10.53
Av. RT = 6.86
} Av. wt = 6.86



CPU Idleness = 0

S.J.F

Av. TAT = 9.53

Av RT = $\frac{7+8.6}{3} = \frac{15.6}{3} = 5.2$

Av. wt = 5.2

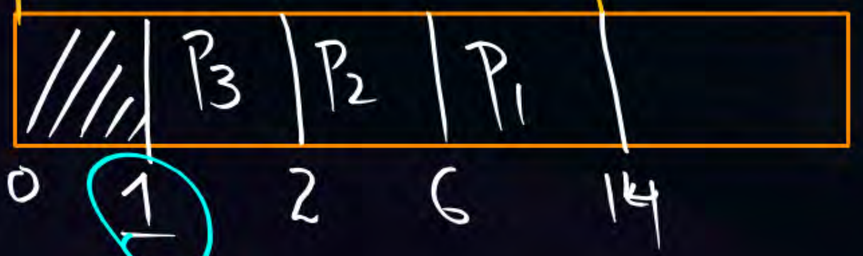
(a) What is the average Turnaround Time and Response Time for these processes with the FCFS scheduling algorithm?

(b) What is the average turnaround time and Response Time for these processes with the SJF Scheduling algorithm?

(c) The SJF algorithm is supposed to improve performance, but notice that we chose to run process P_1 at time 0 because we did not know that two shorter processes would arrive soon. Compute what the average TAT and WT will be if the CPU is left idle for the first 1 unit and then SJF Scheduling is used. Remember that processes P_1 and P_2 are waiting during this idle time, so their waiting time may increase. This algorithm could be known as Future-Knowledge Scheduling. ⟨New-SJF⟩

$Av.R.T = \frac{6 + 1.6 + 0}{3} = \frac{7.6}{3} = 2.5$ ✓

$L = 14$



$Av.TAT = \frac{14 + 5.6 + 1}{3} = \frac{20.6}{3} = 6.8$

$\%CPU\ Idleness = \frac{1}{14}$

$L = Max(CT) - Min(AT)$

$14 - 0 = 14$

Priority Scheduling w/Round-Robin

Hybrid

11:30

- Run the process with the highest priority. Processes with the same Priority run Round-Robin
- Example:

Process	Burst Time	Priority
P_1	4	3 (L)
P_2	5	2
P_3	8	2
P_4	7	1 (H)
P_5	3	3

R.Q: P_1 -- P_5

TAT = 20.8

WT = 15.4

R.T = 4

$= \frac{58}{5} = 11.6$

- Time quantum is 2. Draw the Gantt Chart and Compare Avg. TAT and RT with Pure Round Robin.

CPU

Av. WT = 13.8

Av. TAT = 19.2

Av. RT = $\frac{(20+7+9+0+22)}{5}$

2 mins Summary

Topic One **Priority Scheduling w/Round-Robin**

Topic Two

Topic Three

Topic Four

Topic Five

THANK - YOU