AACS2284 Operating Systems

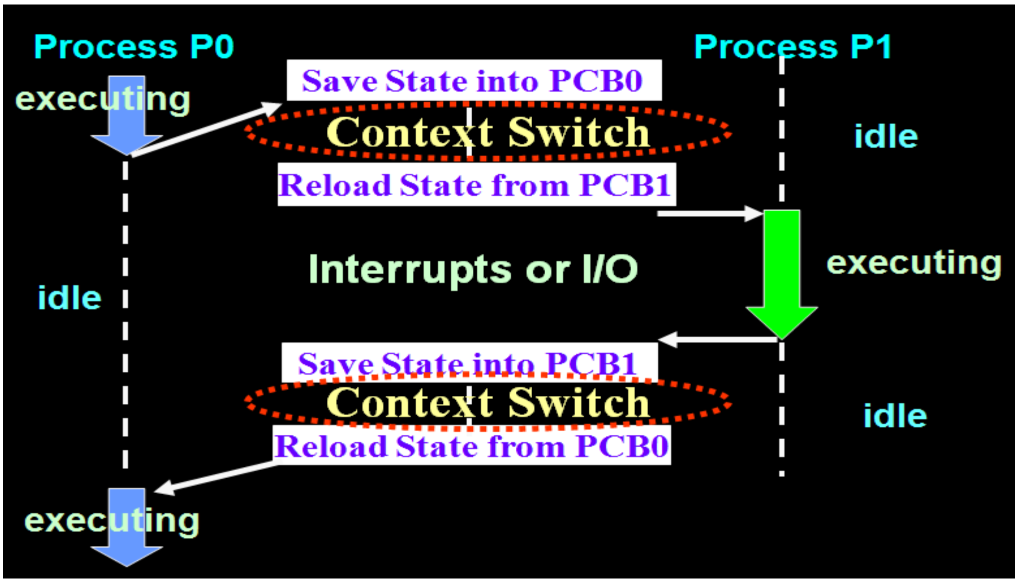
Tutorial 2: Processor Management

Q1. (a) Describe the term “process” in the context of operating systems.

- A process is defined as an entity which represents the basic unit of work to be implemented in the system.
- A process is a ‘task’ or ‘job’ that an individual program has pushed to the CPU (the executable program)
- It is allotted a set of resources (registers, program counter and data section) and will encompass one or more threads.
- Each process in the OS is represented by a PCB repository for information such as Process identification, Process status (HOLD, READY, RUNNING, Waiting), process state etc. PCB’s content varies from process to process

(b) Elaborate how context switch enables multiple processes to share a single processor.

- The job will be placed in process queue based on each job’s characteristics and the jobs will be assigned into CPU to be processed. The job will be moved back to ready queue when the CPU time is finished and assigned to CPU again when the CPU is free. The job will be terminated when it is concluded.



Q2. Differentiate between the followings.

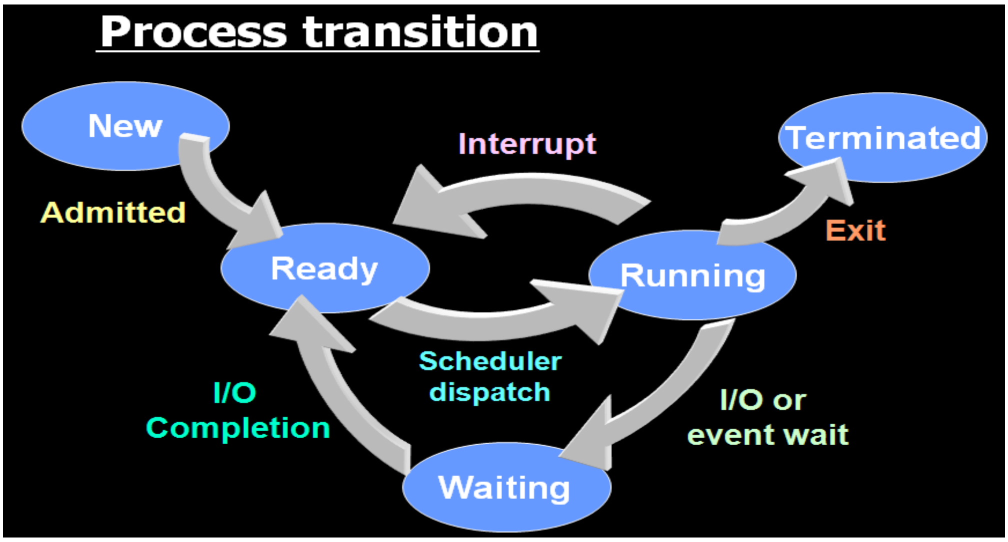
(a) I/O bound and CPU bound

I/O bound	CPU Bound
I/O-bound jobs (such as printing a series of documents) have many brief CPU cycles and long I/O cycles	CPU-bound jobs (such as finding the first 300 prime numbers) have long CPU cycles and shorter I/O cycles.

(b) Preemptive scheduling and non-preemptive scheduling

Preemptive scheduling	Non-preemptive scheduling
Preemptive scheduling policy interrupts processing of a job and transfers the CPU to another job	Once a job captures processor and begins execution, it remains in RUNNING state uninterrupted until it issues an I/O request or until it is finished.

Q3. (a) With the aid of a diagram, explain how a process may switch among the different states. from its creation to termination



- **New** : The process is being created.
- **Running** : Instructions are being executed.
- **Waiting** : The process is waiting for some event to occur. (wait for I/O)
- **Ready** : The process is waiting to be assigned to a CPU
- **Terminated**: The process has finished execution.

(b) Identify the event that will switch a process from running state to ready state and from running state to waiting state.

- **RUNNING to READY**: time interrupt, higher priority process gets ready for execution
- **RUNNING to WAITING**: I/O request, wait for a message or some action from another process.

Q4. Identify the level (short-term, medium-term, long-term) of scheduler that would make a decision on each of the following situation. Justify your answer.

(i) Which temporarily suspended process should be activated in order to balance the process mix?
Medium-term scheduler: determines when processes are to be suspended and resumed.

(ii) Which ready process should be assigned the CPU when it next becomes available?
Short term scheduler (Process Scheduler): Determines which of the ready processes can have CPU resources and for how long.

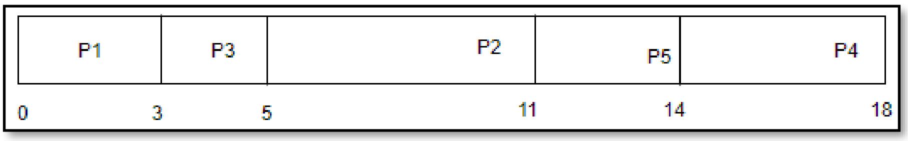
(iii) Which of a series of waiting batch jobs that have been spooled to disk should next be initiated?
Long-term scheduler (Job Scheduler): determines which programs are admitted to the system for execution and when, which ones should be exited.

Q5. (a) Consider the following set of processes, with the length of the CPU burst given in milliseconds (ms). Ignore no context switch time.

Process	Arrival Time	Execution Time	Priority
P1	0	3	1
P2	2	6	2
P3	3	2	3
P4	6	4	3
P5	9	3	4

Draw a **Gantt chart** illustrating the execution of the above processes, and calculate the **average turnaround time** and **average waiting time** based on each of the following algorithms:

(i) Shortest Job First (Non-preemptive)

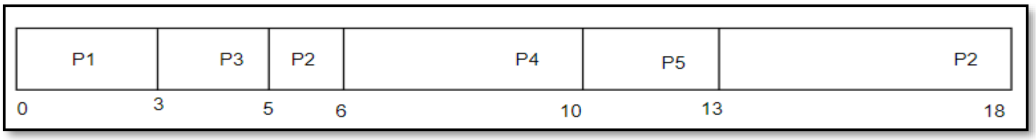


Job	Arrival Time	CPU cycle	Finish Time	Wait time	Turnaround time
P1	0	3	3	0	3
P2	2	6	11	3	9
P3	3	2	5	0	2
P4	6	4	18	8	12
P5	9	3	14	2	5
Average				**2.6**	**6.2**

Average Waiting Time
= ((3-3-0) + (11-6-2) + (5-2-3) + (18-4-6) + (14-3-9)) / 5
= 13 / 5
= 2.6

Turnaround Time
= ((3-0) + (11-2) + (5-3) + (18-6) + (14-9)) / 5
= 31 / 5
= 6.2

(ii) Shortest Remaining Time First (SRTF)

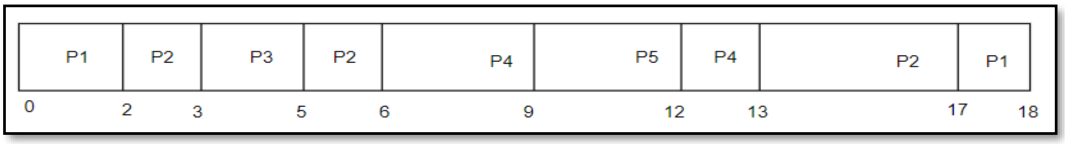


Job	Arrival Time	CPU cycle	Finish Time	Wait time	Turnaround time
P1	0	3	3	0	3
P2	2	6	18	10	16
P3	3	2	5	0	2
P4	6	4	10	0	4
P5	9	3	13	1	4
Average				**2.2**	**5.8**

Average Waiting Time
= ((3-3-0) + (18-6-2) + (5-3-2) + (10-4-6) + (13-3-9)) / 5
= 11 / 5
= 2.2

Turnaround Time
= ((3-0) + (18-2) + (5-3) + (10-6) + (13-9)) / 5
= 29 / 5
= 5.8

(iii) Preemptive Priority (Assume that a larger priority number implies a higher priority)

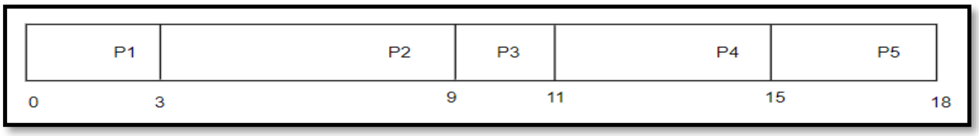


Job	Arrival Time	CPU cycle	Finish Time	Wait time	Turnaround time
P1	0	3	18	15	18
P2	2	6	17	9	15
P3	3	2	5	0	2
P4	6	4	13	3	7
P5	9	3	12	0	3
Average				**5.4**	**9.0**

Average Waiting Time
= ((18-3-0) + (17-6-2) + (5-3-2) + (13-4-6) + (12-3-9)) / 5
= 27 / 5
= 5.4

Turnaround Time
= ((18-0) + (17-2) + (5-2) + (13-6) + (12-9)) / 5
= 45 / 5
= 9.0

(iv) Non-Preemptive Priority (Assume that a smaller priority number implies a higher priority)

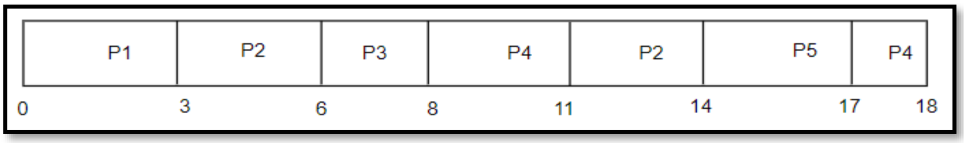


Job	Arrival Time	CPU cycle	Finish Time	Wait time	Turnaround time
P1	0	3	3	0	3
P2	2	6	9	1	7
P3	3	2	11	6	8
P4	6	4	15	5	9
P5	9	3	18	6	9
Average				**3.6**	**7.2**

Average Waiting Time
= ((3-3-0) + (9-6-2) + (11-3-2) + (15-4-6) + (18-3-9)) / 5
= 18 / 5
= 3.6

Turnaround Time
= ((3-0) + (9-2) + (11-3) + (15-6) + (18-9)) / 5
= 36 / 5
= 7.2

(v) Round Robin (Assume that time quantum = 3ms)



Job	Arrival Time	CPU cycle	Finish Time	Wait time	Turnaround time
P1	0	3	3	0	3
P2	2	6	14	6	12
P3	3	2	8	3	5
P4	6	4	18	8	12
P5	9	3	17	5	8
Average				**4.4**	**8.0**

Average Waiting Time
= ((3-3-0) + (14-6-2) + (8-3-2) + (18-4-6) + (17-3-9)) / 5
= 22 / 5
= 4.4

Turnaround Time
= ((3-0) + (14-2) + (8-3) + (18-6) + (17-9)) / 5
= 40 / 5
= 8.0

(b) State why strict non-preemptive scheduling is unlikely to be used in a computer center.
- If non-preemptive scheduling is used in a computer center, a process is capable of keeping other processes waiting for a long time.
- Higher priority processes have to wait the current process finish it's execution and may not to minimize the average waiting time in a computer center.

(c) Is round robin algorithm suitable for interactive systems? State **TWO (2)** reasons to support your answer.
- Yes, Round Robin algorithm is suitable for interactive systems
- It provides reasonable response time to interactive users
- It provides fair CPU allocation

(d) Complete the table below.

Policy	Algorithm	Based on	Pros	Cons
Non-preemptive	First Come First Serve (FCFS)	Arrival time (FIFO)	Batch system	Interactive system
Non-preemptive	Shortest Job First (SJN)	Based on shortest CPU cycle	Job available at the same time	Interactive system
Non-preemptive/ Preemptive	Priority scheduling (PS)	Priority & FCFS	Preferential system	low priority job keeps waiting
Preemptive	Shortest remaining time (SRT)	Based on shortest remaining CPU cycle	Fastest completion	Interactive system
Preemptive	Round robin (RR)	Based on quantum	Equally share CPU	More overhead
Wait Time = Finish Time – CPU Cycle – Arrival Time Turnaround Time = Finish Time – Arrival Time				

Self-Review

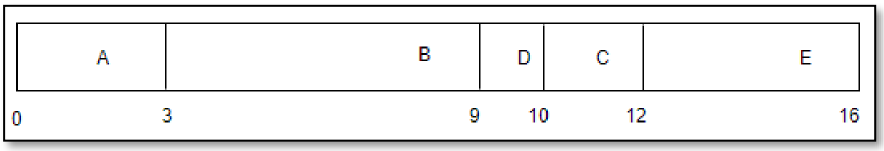
Q1. Consider the list of processes that going to execute as shown in **Table 1:**

Table 1: List of processes going to execute

Process	**Arrival Time** (ms)	**Execution Time** (ms)	**Priority**
A	0	3	3
B	2	6	4
C	5	2	2
D	7	1	1
E	9	4	5

By referring to **Table 1**, draw a timeline for each of the following algorithms and calculate the *Average Waiting Time* and *Average Turnaround Time* for each of the algorithms:

(a) Shortest Job Next (SJN)

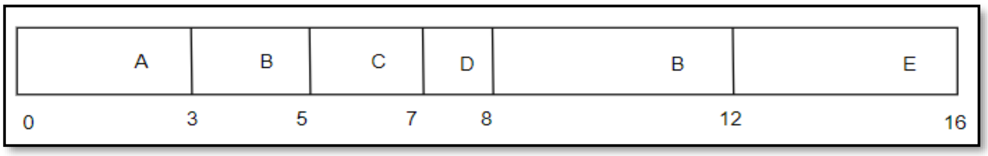


Job	Arrival Time	CPU cycle	Finish Time	Wait time	Turnaround time
A	0	3	3	0	3
B	2	6	9	1	7
C	5	2	12	5	7
D	7	1	10	2	3
E	9	4	16	3	7
Average				**2.2**	**5.4**

Average Waiting Time
= ((3-3-0) + (9-6-2) + (12-5-2) = (10-7-1) + (16-9-4)) / 6
= 11 / 5
= 2.2

Turnaround Time
= ((3-0) + (9-2) + (12-5) + (10-7) + (16-9)) / 5
= 27 / 5
= 5.4

(b) Shortest Remaining Time (SRT)

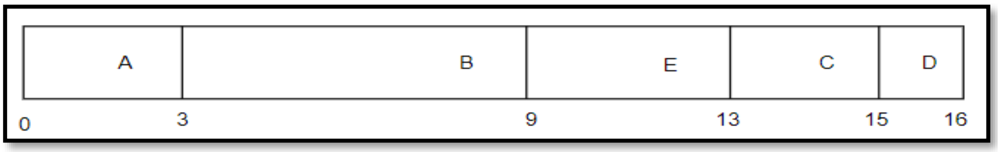


Job	Arrival Time	CPU cycle	Finish Time	Wait time	Turnaround time
A	0	3	3	0	3
B	2	6	12	4	10
C	5	2	7	0	2
D	7	1	8	0	1
E	9	4	16	3	7
Average				**1.4**	**4.6**

Average Waiting Time
= ((3-3-0) + (12-6-2) + (7-2-5) + (8-1-7) + (16-4-9)) / 5
= 7 / 5
= 1. 4

Turnaround Time
= ((3-0) + (12-2) + (7-5) + (8-7) + (16-9)) / 5
= 23 / 5
= 4. 6

(c) Non-Preemptive Priority (Assume that a larger priority implies a higher priority



Job	Arrival Time	CPU cycle	Finish Time	Wait time	Turnaround time
A	0	3	3	0	3
B	2	6	9	1	7
C	5	2	15	8	10
D	7	1	16	8	9
E	9	4	13	0	4
Average				**3.4**	**6.6**

Average Waiting Time
= ((3-3-0) + (9-6-2) + (15-2-5) + (16-1-7) + (13-4-9)) / 5
= 17 / 5
= 3. 4

Turnaround Time
= ((3-0) + (9-2) + (15-5) + (16-7) + (13-9)) / 5
= 33 / 5
= 6.6

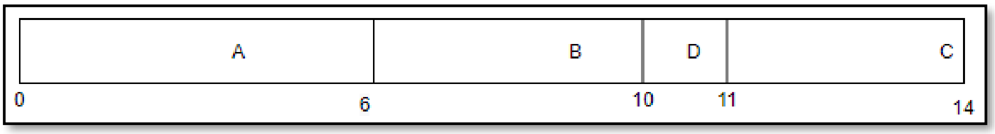
Q2. Consider the list of processes to be performed by processor shown in the **Table 2:**

Table 2: List of processes to be performed by processor

Process	**Arrival Time**	**CPU cycle**
A	0	6
B	1	4
C	7	3
D	9	1

Draw a timeline for each of the following algorithms and calculate the *Average Waiting Time* and *Average Turnaround Time* for each of them:

(a) Shortest Job Next (SJN)

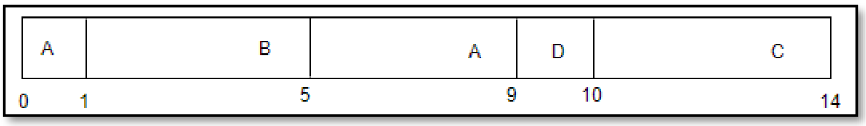


Job	Arrival Time	CPU cycle	Finish Time	Wait time	Turnaround time
A	0	6	6	0	6
B	1	4	10	5	9
C	7	3	14	4	7
D	9	1	11	1	2
Average				**2.5**	**6.0**

Average Waiting Time
= ((6-6-0) + (10-4-1) + (14-3-7) + (11-9-1)) / 4
= 10 / 4
= 2. 5

Turnaround Time
= ((6-0) + (10-1) + (14-7) + (11-9)) / 4
= 24 / 4
= 6.0

(b) Shortest Remaining Time

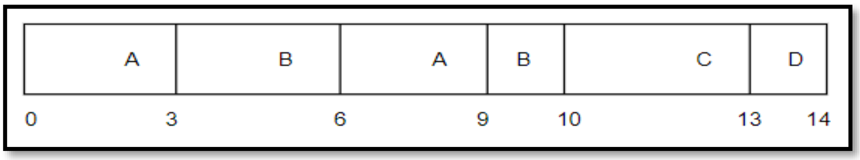


Job	Arrival Time	CPU cycle	Finish Time	Wait time	Turnaround time
A	0	6	10	4	10
B	1	4	5	0	4
C	7	3	14	4	7
D	9	1	11	1	2
Average				**2.25**	**5.75**

Average Waiting Time
= ((10-6-0) + (5-4-1) + (14-3-7) + (11-1-9)) / 4
= 9 / 4
= 2.25

Turnaround Time
= ((10-0) + (5-1) + (14-7) + (11-9)) / 4
= 23 / 4
= 5.75

(c) Round Robin (Assume the time quantum is 3 time slices).



Job	Arrival Time	CPU cycle	Finish Time	Wait time	Turnaround time
A	0	6	9	3	9
B	1	4	10	5	9
C	7	3	13	3	6
D	9	1	14	4	5
Average				**3.75**	**7.25**

Average Waiting Time
= ((9-6-0) + (10-4-1) + (13-3-7) + (14-9-1)) / 4
= 15 / 4
= 3.75

Turnaround Time
= ((9-0) + (10-1) + (13-7) + (14-9)) / 4
= 29 / 4
= 7.25