Vaishak Balachandra

#21

Homework #1 (Due date: 2/21/2025)

1. Below is an algorithm for Bounded Buffer problem. Fill in the table of the values of semaphores. (2.5 points)

```
Shared data
semaphore full, empty, mutex;
Initially:
full = 0, empty = n, mutex = 1
Producer process:
do {
                    …
                produce an item in nextp
                    …
                wait(empty);
                wait(mutex);
                    …
                add nextp to buffer
                    …
                signal(mutex);
                signal(full);
            } while (1);
Consumer process:
do {
                wait(full)
                wait(mutex);
                    …
                remove an item from buffer to nextc
                    …
                signal(mutex);
                signal(empty);
                    …
                consume the item in nextc
                    …
            } while (1);
```

events	mutex	full	empty
produce 1 item	1 → 0 → 1	0 → 1	n → n-1
produce 1 item	1 → 0 → 1	1 → 2	n-1 → n-2
produce 1 item	1 → 0 → 1	2 → 3	n-2 → n-3
consume 1 item	1 → 0 → 1	3 → 2	n-3 → n-2
consume 1 item	1 → 0 → 1	2 → 1	n-2 → n-1
produce 1 item	1 → 0 → 1	1 → 2	n-1 → n-2
consume 1 item	1 → 0 → 1	2 → 1	n-2 → n-1
consume 1 item	1 → 0 → 1	1 → 0	n-1 → n.

For n = 3 :

events	mutex	full	empty
produce 1 item	1 → 0 → 1	0 → 1	3 → 2
produce 1 item	1 → 0 → 1	1 → 2	2 → 1
produce 1 item	1 → 0 → 1	2 → 3	1 → 0
consume 1 item	1 → 0 → 1	3 → 2	0 → 1
consume 1 item	1 → 0 → 1	2 → 1	1 → 2
produce 1 item	1 → 0 → 1	1 → 2	2 → 1
consume 1 item	1 → 0 → 1	2 → 1	1 → 2
consume 1 item	1 → 0 → 1	1 → 0	2 → 3

2. Below is an algorithm for Readers-Writers problem. Fill in the table of the values of semaphores or variables. (2.5 points)

Shared data
semaphore mutex, db;
Initially
mutex = 1, db = 1, readcount = 0

Writer process:

```
wait(db);
        ...
    writing is performed
        ...
signal(db);
```

Reader process:

```
wait(mutex);
readcount++;
if (readcount == 1)
        wait(db);
signal(mutex);
        ...
    reading is performed
        ...
wait(mutex);
readcount--;
if (readcount == 0)
        signal(db);
signal(mutex):
```

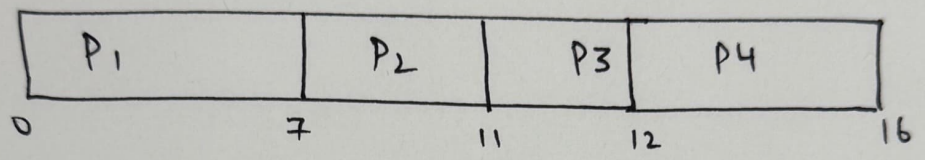
event	mutex = 1	db = 1	readcount = 0
a writer comes	1	1 → 0	0
a reader comes	1 → 0	0 → -1	0 → 1
a writer has done	0 → 1	-1 → 0	1
a reader comes	1 → 0 → 1	0	1 → 2
a reader comes	1 → 0 → 1	0	2 → 3
a writer comes	1	0 → -1	3
a reader has done	1 → 0 → 1	-1	3 → 2
a reader has done	1 → 0 → 1	-1	2 → 1
a reader has done	1 → 0 → 1	-1 → 0	1 → 0
a writer comes	1	0 → -1	0

3. Consider the following set of processes, with length of the Burst Time given in milliseconds: (5 points)

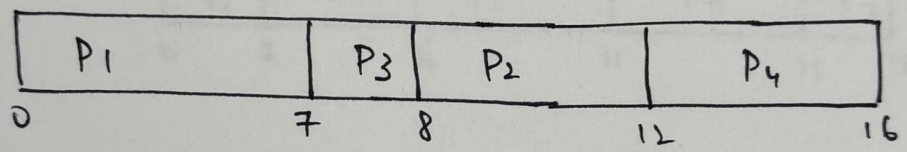
Process	Arrival Time	Burst Time	Priority
P1	0.0	7	2
P2	2.0	4	1
P3	4.0	1	3
P4	5.0	4	2

1) Draw Gantt charts **with ready queues** if applicable, that using FCFS, SJF, Priority (a smaller priority number implies a higher priority), and RR (quantum = 1), with non-preemptive for all the algorithms applicable.

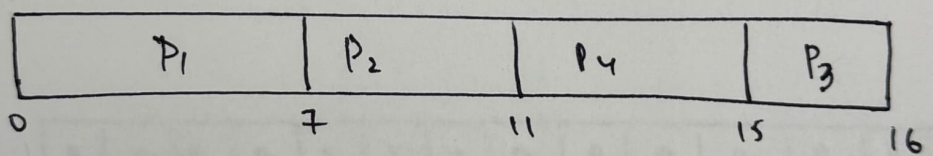
FCFS :-



SJF :-



Priority :-



Round Robin : → RR is an example for pre-emptive algorithm. Hence ignored !!

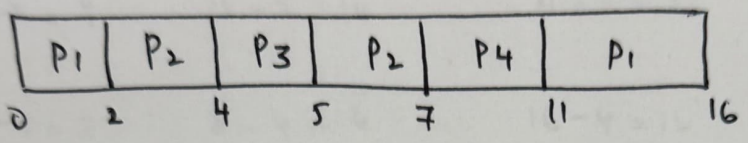
→ As it is always pre-emptive.

2) Repeat 1) with preemptive for the **applicable algorithms only**.

FCFS :- ⟶ is always non-preemptive

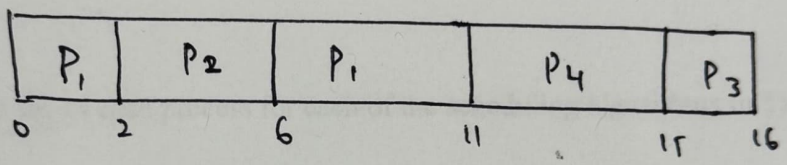
⟶ Thus ignored Here !!

SJF Preemptive :-



↳ uses AT & BT

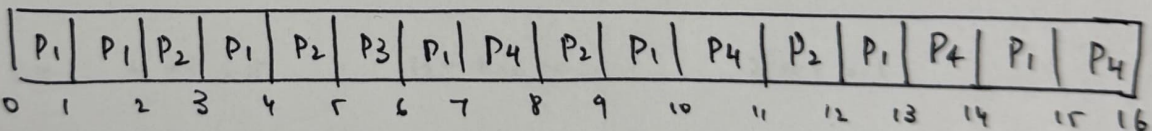
Priority Preemptive :



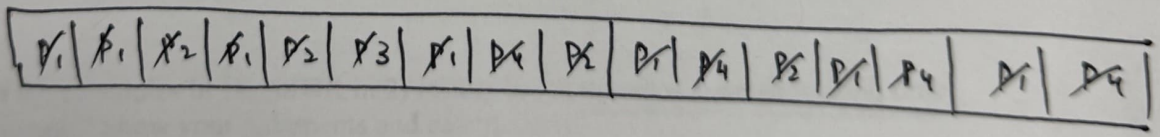
↳ uses
- Priority
- AT

Round Robin (RR) :

↳ Gantt chart :



↳ Ready Queue :



3) What is the Turnaround time of each process for each of the scheduling algorithms in 1) and RR in 2)?

	FCFS	SJF	Priority	RR
P1	7-0 = 7	7 - 0 = 7	7 - 0 = 7	15 - 0 = 15
P2	11 -2 = 9	12 - 2 = 10	11 - 2 = 9	12 - 2 = 10
P3	12 -4 = 8	8 - 4 = 4	16 - 4 = 12	6 - 4 = 2
P4	16 - 5 = 11	16 - 5 = 11	15 - 5 = 10	16 - 5 = 11

4) What is the Waiting time of each process for each of the scheduling algorithms in 1) and RR in 2).

	FCFS	SJF	Priority	RR
P1	7 - 7 = 0	7 - 7 = 0	7 - 7 = 0	15 - 7 = 8
P2	9 - 4 = 5	10 - 4 = 6	9 - 4 = 5	10 - 4 = 6
P3	8 - 1 = 7	4 - 1 = 3	12 - 1 = 11	2 - 1 = 1
P4	11 - 4 = 7	11 - 4 = 7	10 - 4 = 6	11 - 4 = 7

5) Which of the schedules in 1) and RR in 2) results in the minimal average waiting time over all processes? Show your judgments and calculations.

	FCFS	SJF	Priority	RR
Avg wait Time	$\frac{0+5+7+7}{4}$	$\frac{0+6+3+7}{4}$	$\frac{0+5+11+6}{4}$	$\frac{8+6+1+7}{4}$
	= 4.7ms	= 4ms	= 5.5ms	= 5.5ms

Conclusion / Judgement : SJF has the lowest average waiting time.