National University of Modern Languages

Department of Computer Sciences

Subject: Data Structures and Algorithm	**Instructor:** Zainab Malik
Lab Assignment: 2	**Class:** BSCS 3BE
Student Roll No:2883	**Muhammad Awais**

Assigned Task:

Q1. Study following process scheduling techniques and suggest appropriate data structures for their implementation along with proper justification/reason. (CLO-3, Marks 15)

A program that invoked for execution is known as a process. For execution, process enters a queue first (i.e. READY queue) and from this READY queue, process gets dequeued and utilize CPU for the execution of respective task. Process may have different attributes such as process ID, priority, arrival time, completion time, execution time, wait time and turn-around time.

- **Process ID**: Unique identifier that differentiate one process from the other. e.g. p1, p2 and p3 etc.
- **Arrival Time**: Time at which the process arrives in the ready queue.
- **Completion Time**: Time at which process completes its execution.
- **Process Execution Time**: Time a process takes for executing its task.
- **Turn Around Time:** Time difference between completion time and arrival time
- **Process Wait Time/Service Time**: Time a process waits in queue due to the execution of other process in the queue or Time difference between turnaround time and execution time.
- **Priority**: Priority is associated with each process that determines the order of execution in *priority scheduling only*.

  For understanding, different scheduling techniques, consider following set of processes along with the associated priority, arrival and execution time *(provided by the user).*

Table 1: List of Processes

Process ID	Priority	Arrival Time	Execution Time (milli sec)	Wait Time (milli sec)	Completion Time	Turnaround Time
P0	2	0	5			
P1	0	1	3			
P2	3	2	8			
P3	1	3	6			

1. **First-Come-First-Served Scheduling**

   As name suggests that the process coming first should be served/ or executed first. e.g. there are4 process P0, P1, P2 and P3. According to the arrival time (mentioned in the table 1), p0 came first, therefore it should be executed first for 5 sec as per its execution time. As it was the first oneso, its service time/wait time is 0. P1 came at 1 (arrival time), so, it should be executed at secondnumber for 3 sec as per the execution time provided by the user. As P1 was executed after P0 whose execution time was 5 sec, that means P1 waited for 5sec before it execution. Similarly, p3,will be executed at the end therefore, its waiting time should be 5+3+8=16 sec.

Process ID	Priority	Arrival Time	Execution Time(milli sec)	Wait Time (milli sec)	CompletionTime	TurnaroundTime
P0	2	0	5	0	5	5
P1	0	1	3	5	8	7
P2	3	2	8	8	16	14
P3	1	3	6	16	22	19

Suggested Data Structure and why?

To achieved the **First come first served** behavior in the processing of above scheduling we used a **Linked list base Queue.**

Justification:
Because to achieved FCFS implementation we follow the Queue which allows to execute the process in First come First Served.

2. Shortest-Job-First Scheduling

The process with shortest execution time should be served/ or executed first. From table 1, as perthe execution time process should be executed in following sequence: p1, p0, p3 and p2. However, wait time, completion time and turnaround time should be calculated accordingly.

Suggested Data Structure and why?

To achieved the **Shortest Job First** implementation in the processing of above scheduling we used a **Priority Queue.**
we
Justification:

Shortest Job First algorithm is one of scheduling technique which allow the **shortest burst time (Takes shortest time to execute)** process to execute first. So, to implement **SJF** we need to priorities the tasks with burst time. To implement the **SJF** we used **Priority Queue. Here** the **highest priority the one which burst time is shortest / minimum** than all and follow this rule till end.

3. Longest-Job-First Scheduling

The process with longest execution time should be served/ or executed first. From table 1, as perthe execution time process should be executed in following sequence: p2, p3, p0 and p1. However, wait time, completion time and turnaround time should be calculated accordingly.

Suggested Data Structure and why?

To achieved the **Longest Job First** implementation in the processing of above scheduling we used a **Priority Queue.**
we
Justification:

Longest Job First algorithm is one of scheduling technique which allow the **longest burst time (Takes highest time to execute)** process to execute first. So, to implement **LJF** we need to priorities the tasks with **burst time**. To implement the **LJF** we used **Priority Queue. Here** the **highest priority the one which burst time is longest / maximum** than all and follow this rule till end.

4. **Priority Scheduling**
   The process with highest Priority should be processed first/ or executed first. For example from table 1, as per the associated priorities process should be executed in following sequence: p1, p3,p0 and p2. However, wait time, completion time and turnaround time should be calculated accordingly.

Suggested Data Structure and why?

To achieved the **Priority Scheduling** implementation, we used a **Priority Queue.**

Justification:

Priority algorithm is one of scheduling technique which allow the **highest priority (here Highest priority means smallest in number integer)** process to execute first.

5. **Round Robin Scheduling** Each process is provided with a fix time to execute; it is called a quantum. Once a process is executed for a given time period, it re-enter the ready **queue** (if it is incomplete) with reduced execution time and again wait for it turns. The wait time, completion time and turnaround time should be calculated accordingly. (explore more over the internet)

Suggested Data Structure and why?

To implement **Round Robin Scheduling**, we used a **circular Queue.**

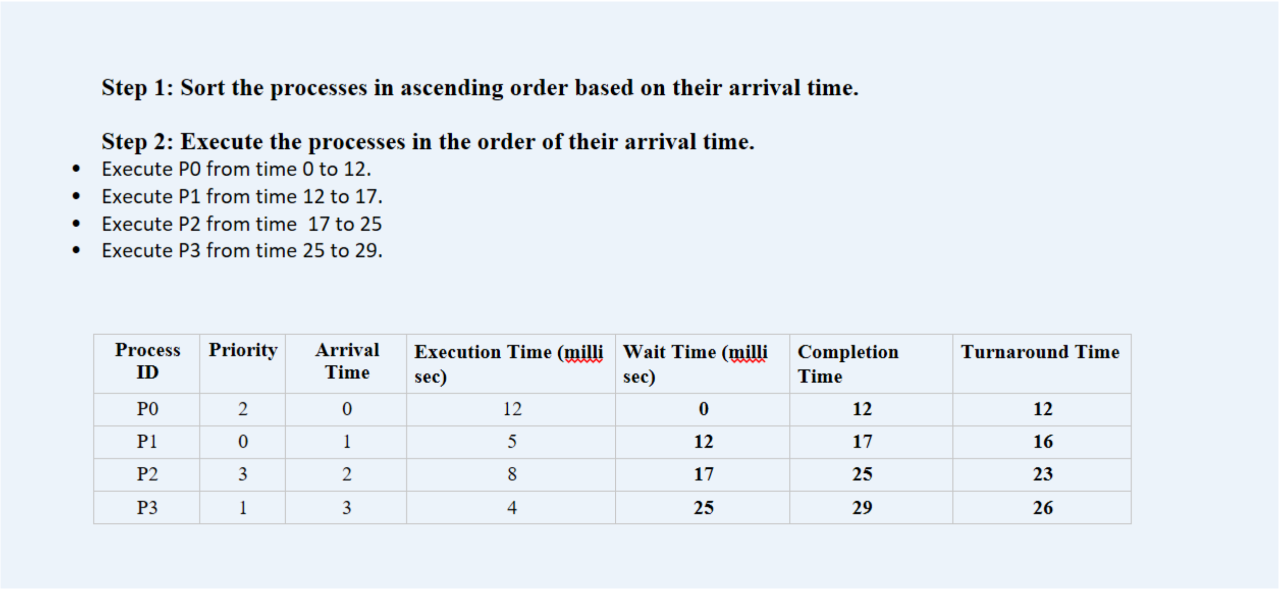
Justification:

Round Robin Scheduling algorithm is one of scheduling technique which follow First come First Served under a time Quantum **(A specific time which share the execution process among tasks. This defined time is called Time Quantum).** If the task does not complete in the **time Quantum** than the process will **interrupted by CPU and proceed next in the Queue.** Once all tasks in Queue gets a given time period to execute incomplete tasks will execute again with **Time Quantum**, So the process work in a **circular behavior**.

Q2. Demonstrate the working of suggested data structure for each scenario using the followinginput. For round robin you may consider quantum time as 3 sec. (CLO-2, Marks 25)

Process ID	Priority	Arrival Time	Execution Time(milli sec)	Wait Time (milli sec)	CompletionTime	TurnaroundTime
P0	2	0	12sec			
P1	0	1	5sec			
P2	3	2	8sec			
P3	1	3	4sec			

First come first served

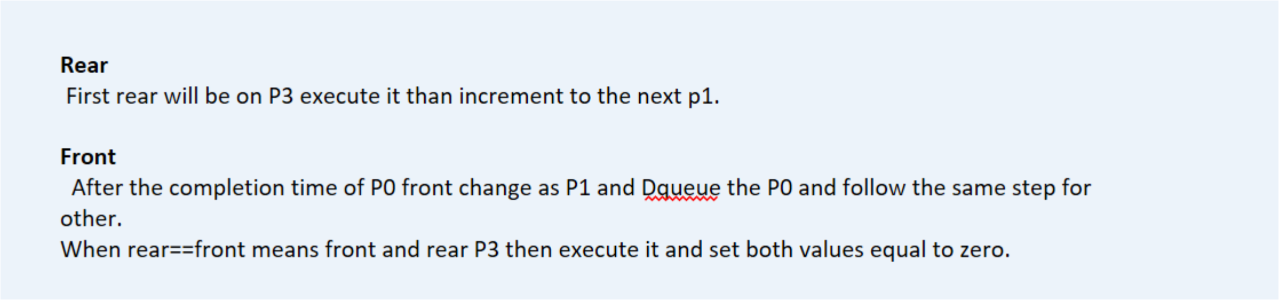


Step 1: Sort the processes in ascending order based on their arrival time.

Step 2: Execute the processes in the order of their arrival time.

- Execute P0 from time 0 to 12.
- Execute P1 from time 12 to 17.
- Execute P2 from time 17 to 25
- Execute P3 from time 25 to 29.

Process ID	Priority	Arrival Time	Execution Time (milli sec)	Wait Time (milli sec)	Completion Time	Turnaround Time
P0	2	0	12	**0**	**12**	**12**
P1	0	1	5	**12**	**17**	**16**
P2	3	2	8	**17**	**25**	**23**
P3	1	3	4	**25**	**29**	**26**



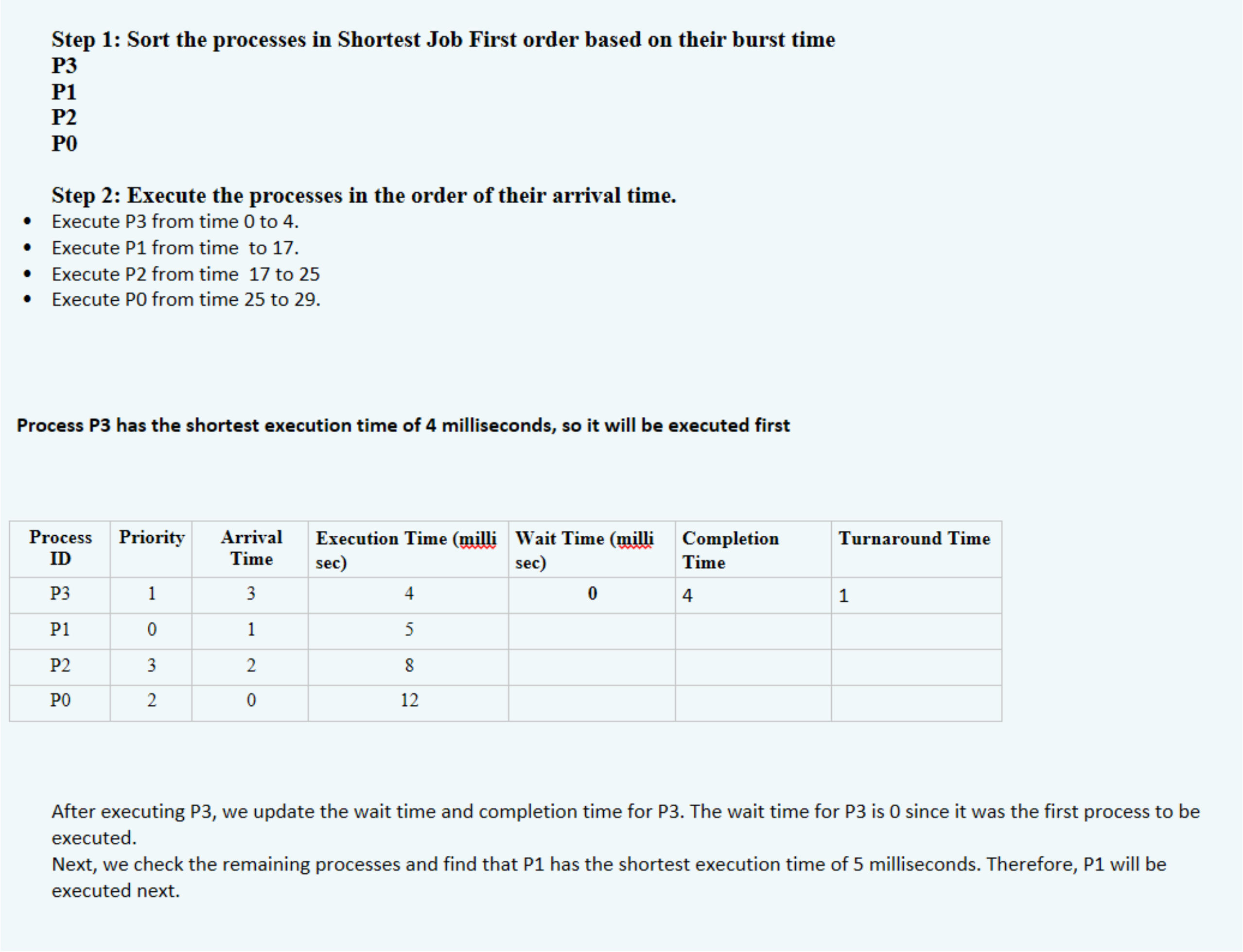
Rear

First rear will be on P3 execute it than increment to the next p1.

Front

After the completion time of P0 front change as P1 and Dqueue the P0 and follow the same step for other.
When rear==front means front and rear P3 then execute it and set both values equal to zero.

Shortest Job First



Step 1: Sort the processes in Shortest Job First order based on their burst time
P3
P1
P2
P0

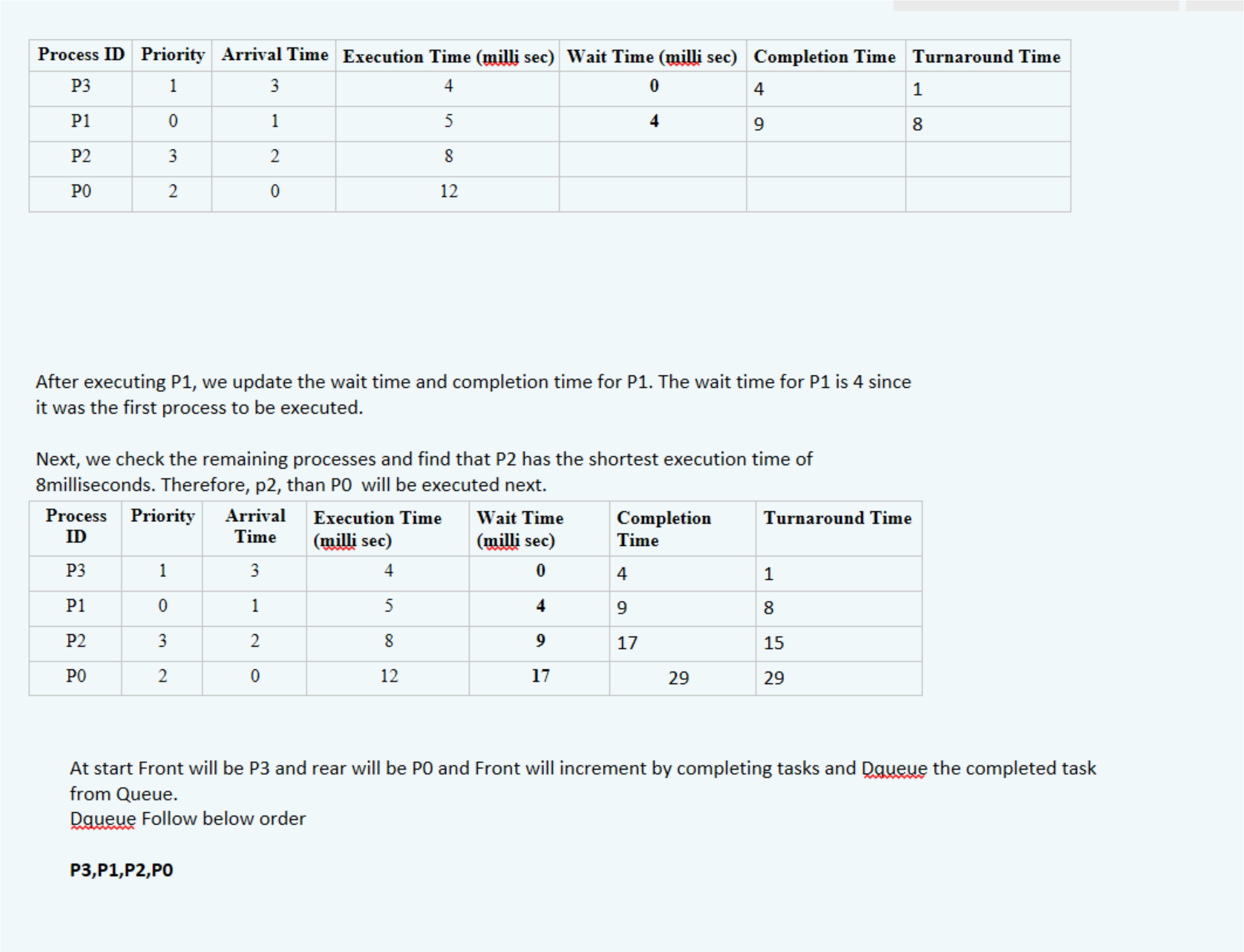
Step 2: Execute the processes in the order of their arrival time.

- Execute P3 from time 0 to 4.
- Execute P1 from time to 17.
- Execute P2 from time 17 to 25
- Execute P0 from time 25 to 29.

Process P3 has the shortest execution time of 4 milliseconds, so it will be executed first

Process ID	Priority	Arrival Time	Execution Time (milli sec)	Wait Time (milli sec)	Completion Time	Turnaround Time
P3	1	3	4	**0**	4	1
P1	0	1	5			
P2	3	2	8			
P0	2	0	12			

After executing P3, we update the wait time and completion time for P3. The wait time for P3 is 0 since it was the first process to be executed.
Next, we check the remaining processes and find that P1 has the shortest execution time of 5 milliseconds. Therefore, P1 will be executed next.



Process ID	Priority	Arrival Time	Execution Time (milli sec)	Wait Time (milli sec)	Completion Time	Turnaround Time
P3	1	3	4	**0**	4	1
P1	0	1	5	**4**	9	8
P2	3	2	8			
P0	2	0	12			

After executing P1, we update the wait time and completion time for P1. The wait time for P1 is 4 since it was the first process to be executed.

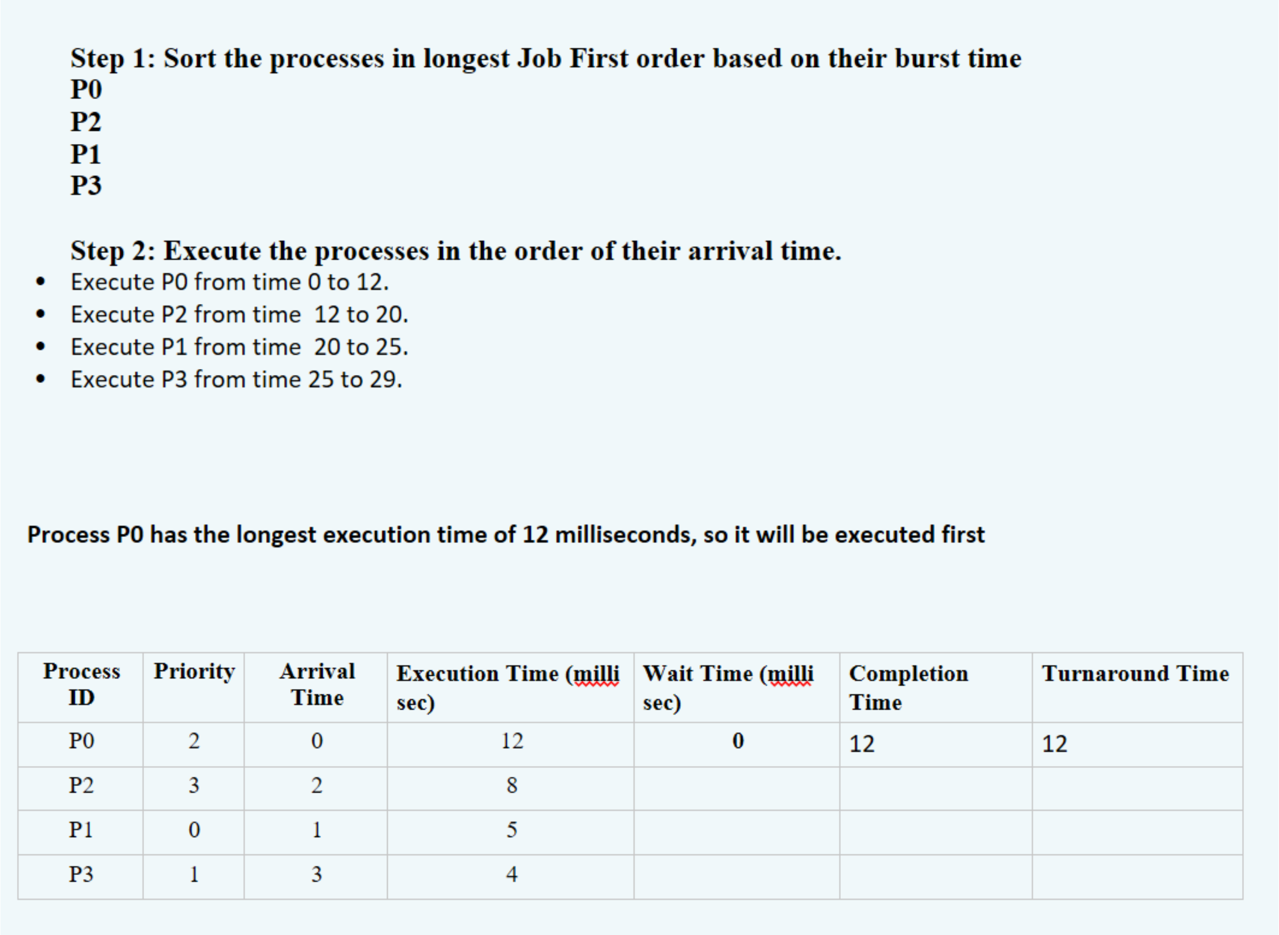
Next, we check the remaining processes and find that P2 has the shortest execution time of 8milliseconds. Therefore, p2, than P0 will be executed next.

Process ID	Priority	Arrival Time	Execution Time (milli sec)	Wait Time (milli sec)	Completion Time	Turnaround Time
P3	1	3	4	**0**	4	1
P1	0	1	5	**4**	9	8
P2	3	2	8	**9**	17	15
P0	2	0	12	**17**	29	29

At start Front will be P3 and rear will be P0 and Front will increment by completing tasks and Dqueue the completed task from Queue.
Dqueue Follow below order

P3,P1,P2,P0

Longest Job First



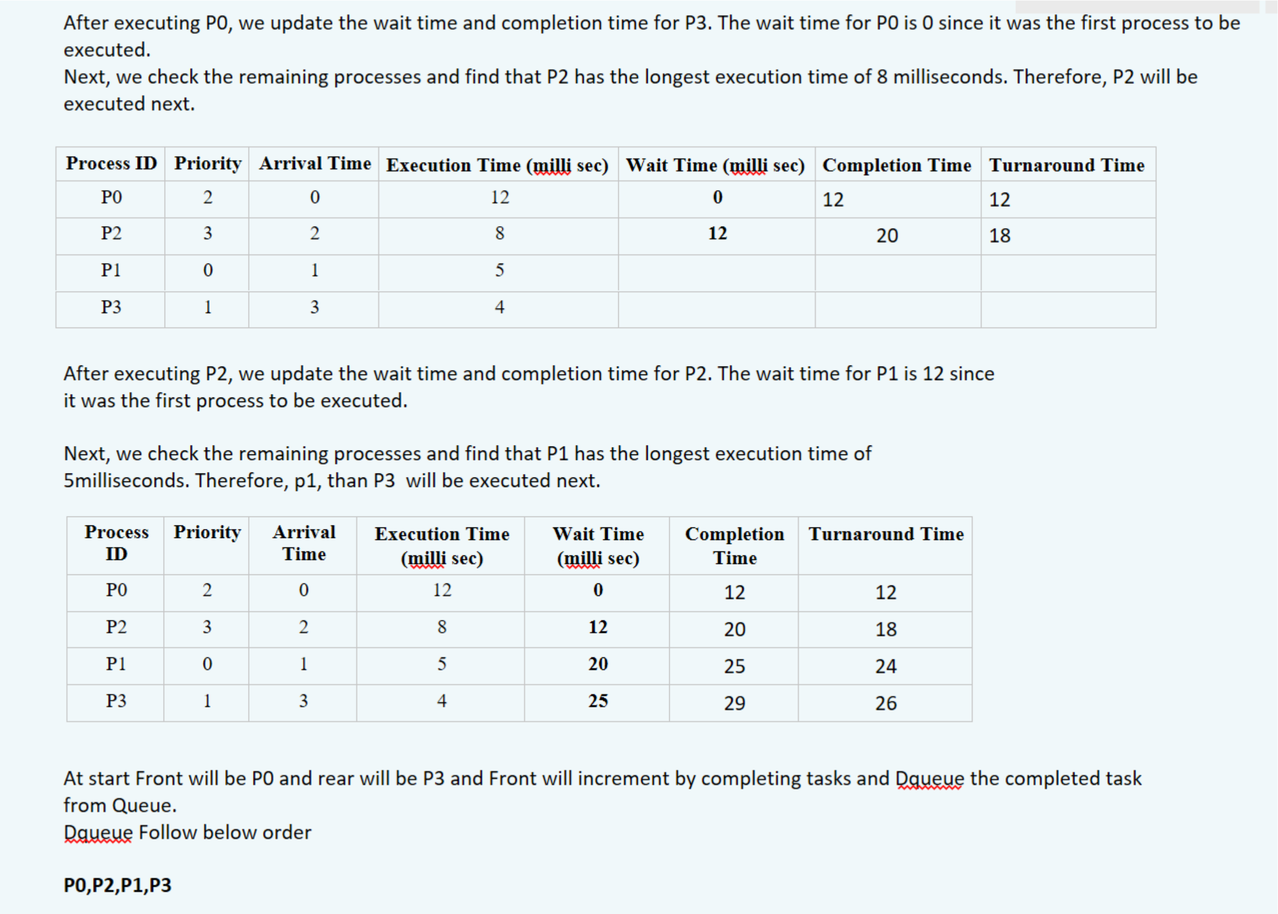
Step 1: Sort the processes in longest Job First order based on their burst time
P0
P2
P1
P3

Step 2: Execute the processes in the order of their arrival time.

- Execute P0 from time 0 to 12.
- Execute P2 from time 12 to 20.
- Execute P1 from time 20 to 25.
- Execute P3 from time 25 to 29.

Process P0 has the longest execution time of 12 milliseconds, so it will be executed first

Process ID	Priority	Arrival Time	Execution Time (milli sec)	Wait Time (milli sec)	Completion Time	Turnaround Time
P0	2	0	12	**0**	12	12
P2	3	2	8			
P1	0	1	5			
P3	1	3	4			



After executing P0, we update the wait time and completion time for P3. The wait time for P0 is 0 since it was the first process to be executed.
Next, we check the remaining processes and find that P2 has the longest execution time of 8 milliseconds. Therefore, P2 will be executed next.

Process ID	Priority	Arrival Time	Execution Time (milli sec)	Wait Time (milli sec)	Completion Time	Turnaround Time
P0	2	0	12	**0**	12	12
P2	3	2	8	**12**	20	18
P1	0	1	5			
P3	1	3	4			

After executing P2, we update the wait time and completion time for P2. The wait time for P1 is 12 since it was the first process to be executed.

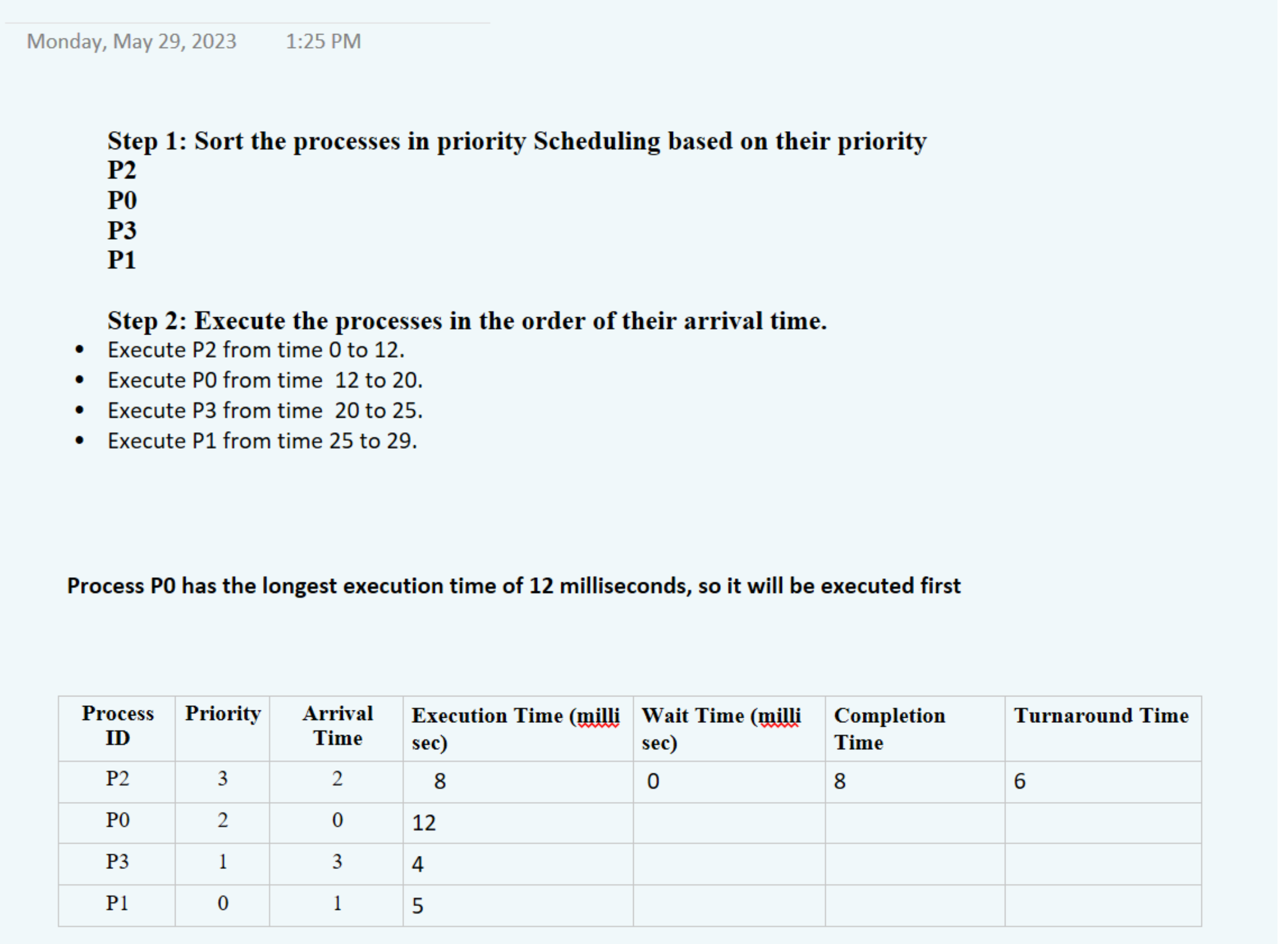
Next, we check the remaining processes and find that P1 has the longest execution time of 5milliseconds. Therefore, p1, than P3 will be executed next.

Process ID	Priority	Arrival Time	Execution Time (milli sec)	Wait Time (milli sec)	Completion Time	Turnaround Time
P0	2	0	12	**0**	12	12
P2	3	2	8	**12**	20	18
P1	0	1	5	**20**	25	24
P3	1	3	4	**25**	29	26

At start Front will be P0 and rear will be P3 and Front will increment by completing tasks and Dqueue the completed task from Queue.
Dqueue Follow below order

P0,P2,P1,P3

Priority Scheduling



Monday, May 29, 2023 1:25 PM

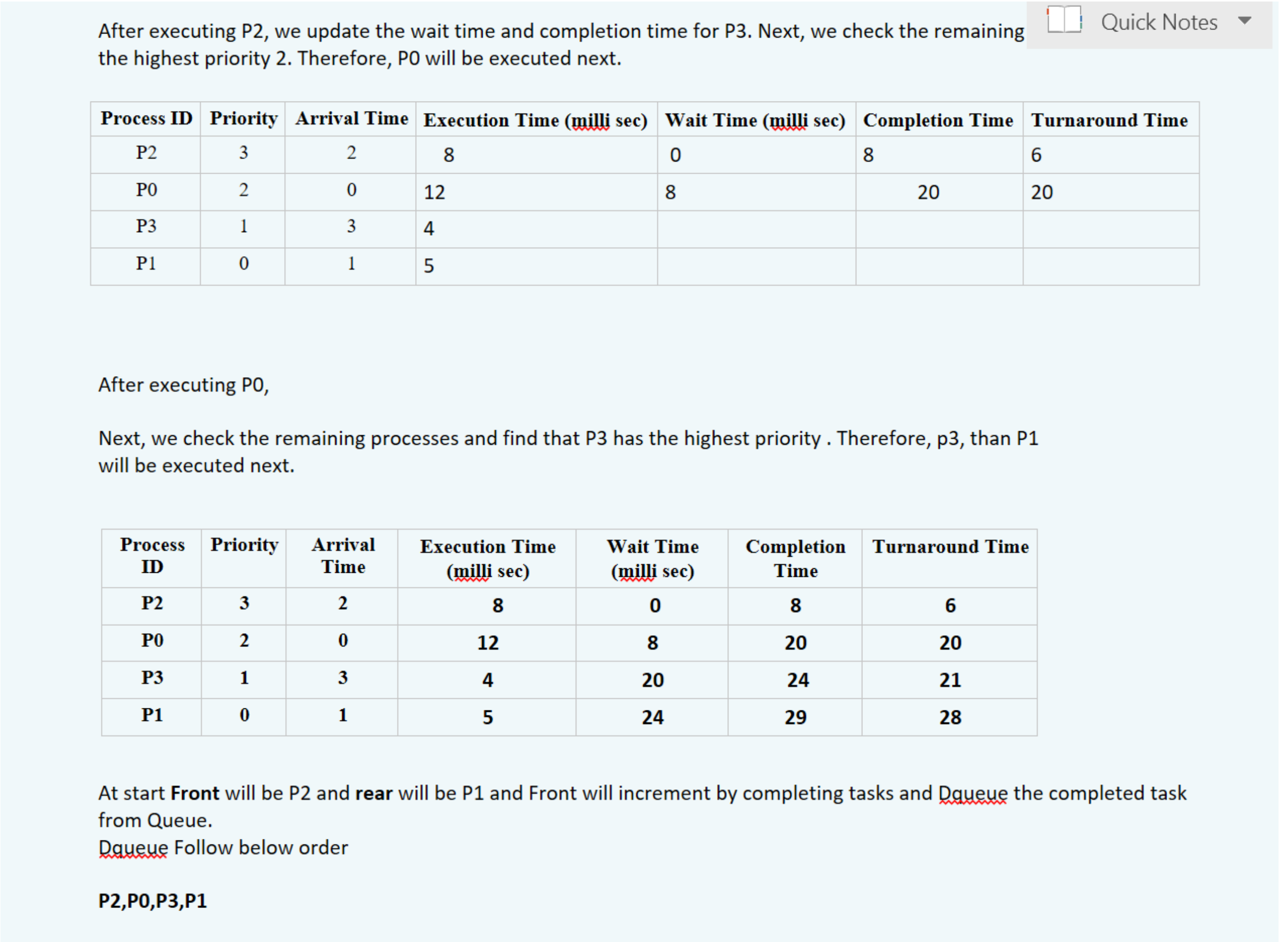
Step 1: Sort the processes in priority Scheduling based on their priority
P2
P0
P3
P1

Step 2: Execute the processes in the order of their arrival time.

- Execute P2 from time 0 to 12.
- Execute P0 from time 12 to 20.
- Execute P3 from time 20 to 25.
- Execute P1 from time 25 to 29.

Process P0 has the longest execution time of 12 milliseconds, so it will be executed first

Process ID	Priority	Arrival Time	Execution Time (milli sec)	Wait Time (milli sec)	Completion Time	Turnaround Time
P2	3	2	8	0	8	6
P0	2	0	12			
P3	1	3	4			
P1	0	1	5			



After executing P2, we update the wait time and completion time for P3. Next, we check the remaining
the highest priority 2. Therefore, P0 will be executed next.

Process ID	Priority	Arrival Time	Execution Time (milli sec)	Wait Time (milli sec)	Completion Time	Turnaround Time
P2	3	2	8	0	8	6
P0	2	0	12	8	20	20
P3	1	3	4			
P1	0	1	5			

After executing P0,

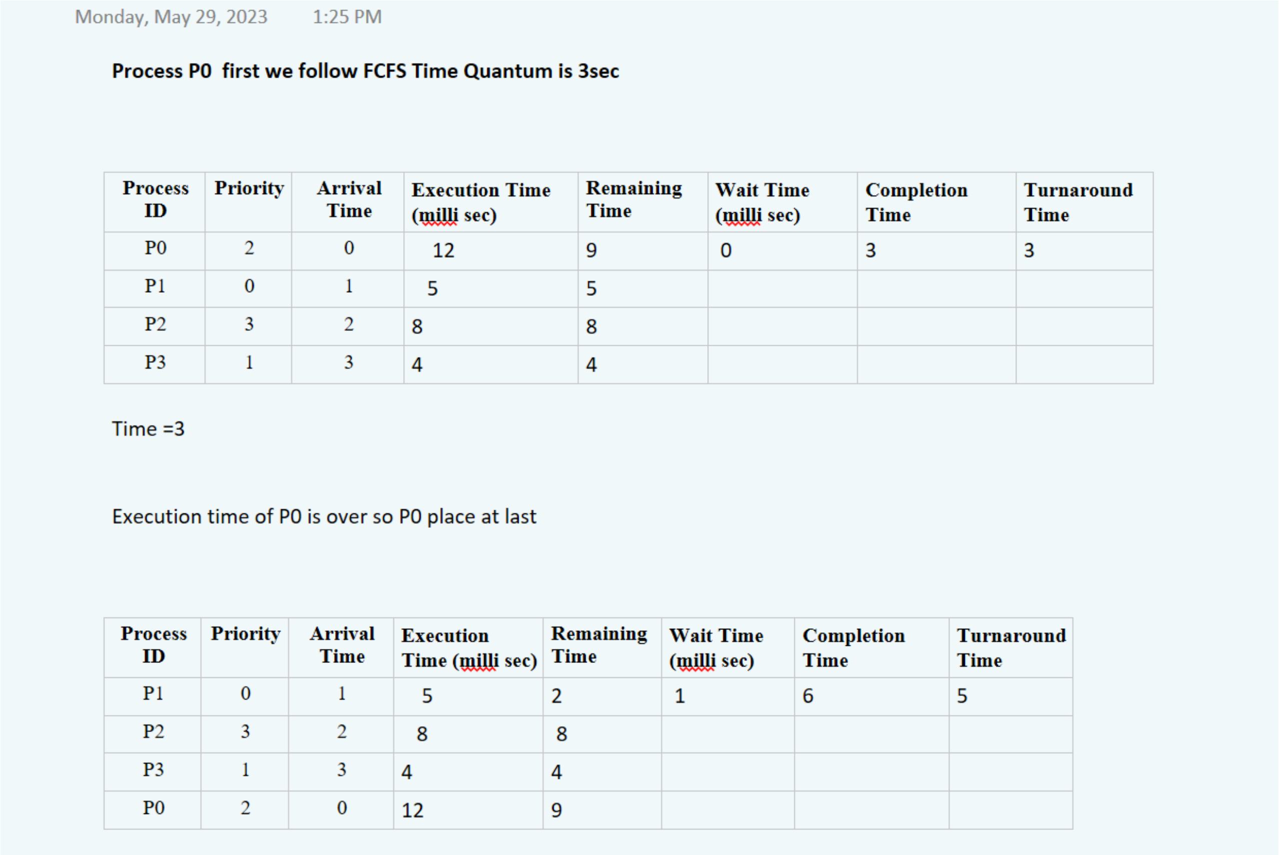
Next, we check the remaining processes and find that P3 has the highest priority . Therefore, p3, than P1 will be executed next.

Process ID	Priority	Arrival Time	Execution Time (milli sec)	Wait Time (milli sec)	Completion Time	Turnaround Time
P2	**3**	**2**	**8**	**0**	**8**	**6**
P0	**2**	**0**	**12**	**8**	**20**	**20**
P3	**1**	**3**	**4**	**20**	**24**	**21**
P1	**0**	**1**	**5**	**24**	**29**	**28**

At start **Front** will be P2 and **rear** will be P1 and Front will increment by completing tasks and Dqueue the completed task from Queue.
Dqueue Follow below order

P2,P0,P3,P1

Round Robin Scheduling



Monday, May 29, 2023 1:25 PM

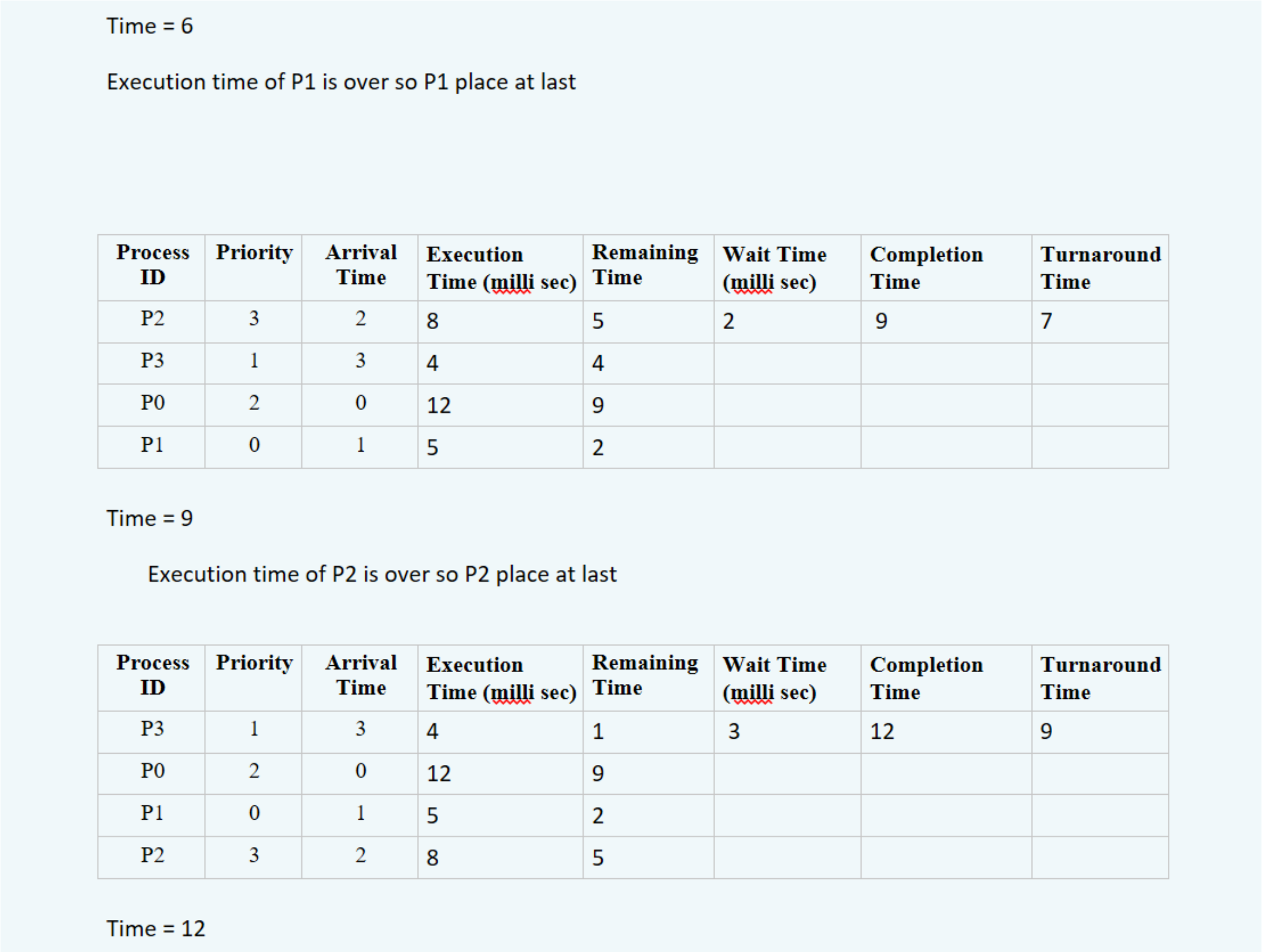
Process P0 first we follow FCFS Time Quantum is 3sec

Process ID	Priority	Arrival Time	Execution Time (milli sec)	Remaining Time	Wait Time (milli sec)	Completion Time	Turnaround Time
P0	2	0	12	9	0	3	3
P1	0	1	5	5			
P2	3	2	8	8			
P3	1	3	4	4			

Time =3

Execution time of P0 is over so P0 place at last

Process ID	Priority	Arrival Time	Execution Time (milli sec)	Remaining Time	Wait Time (milli sec)	Completion Time	Turnaround Time
P1	0	1	5	2	1	6	5
P2	3	2	8	8			
P3	1	3	4	4			
P0	2	0	12	9			



Time = 6

Execution time of P1 is over so P1 place at last

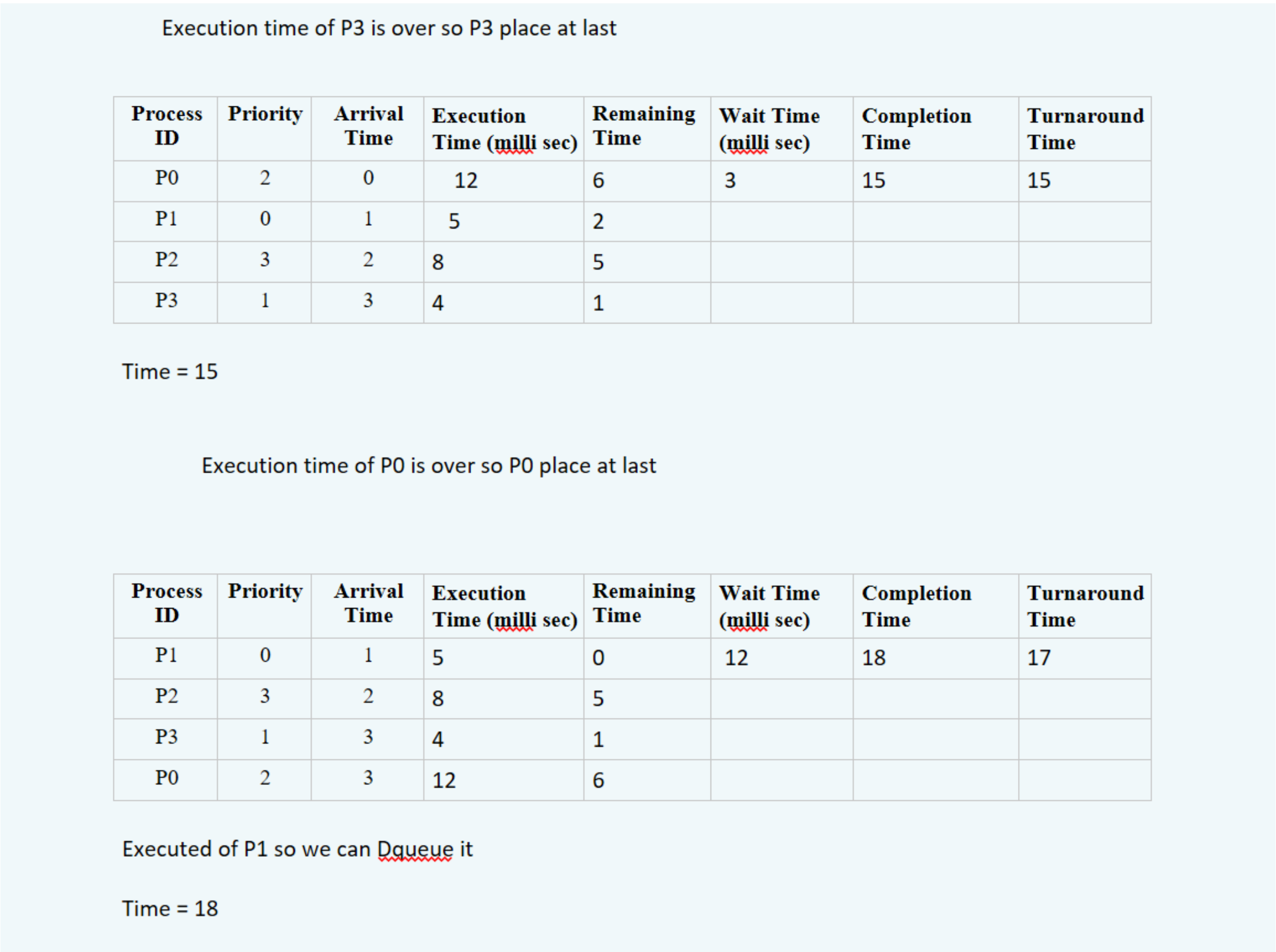
Process ID	Priority	Arrival Time	Execution Time (milli sec)	Remaining Time	Wait Time (milli sec)	Completion Time	Turnaround Time
P2	3	2	8	5	2	9	7
P3	1	3	4	4			
P0	2	0	12	9			
P1	0	1	5	2			

Time = 9

Execution time of P2 is over so P2 place at last

Process ID	Priority	Arrival Time	Execution Time (milli sec)	Remaining Time	Wait Time (milli sec)	Completion Time	Turnaround Time
P3	1	3	4	1	3	12	9
P0	2	0	12	9			
P1	0	1	5	2			
P2	3	2	8	5			

Time = 12



Execution time of P3 is over so P3 place at last

Process ID	Priority	Arrival Time	Execution Time (milli sec)	Remaining Time	Wait Time (milli sec)	Completion Time	Turnaround Time
P0	2	0	12	6	3	15	15
P1	0	1	5	2			
P2	3	2	8	5			
P3	1	3	4	1			

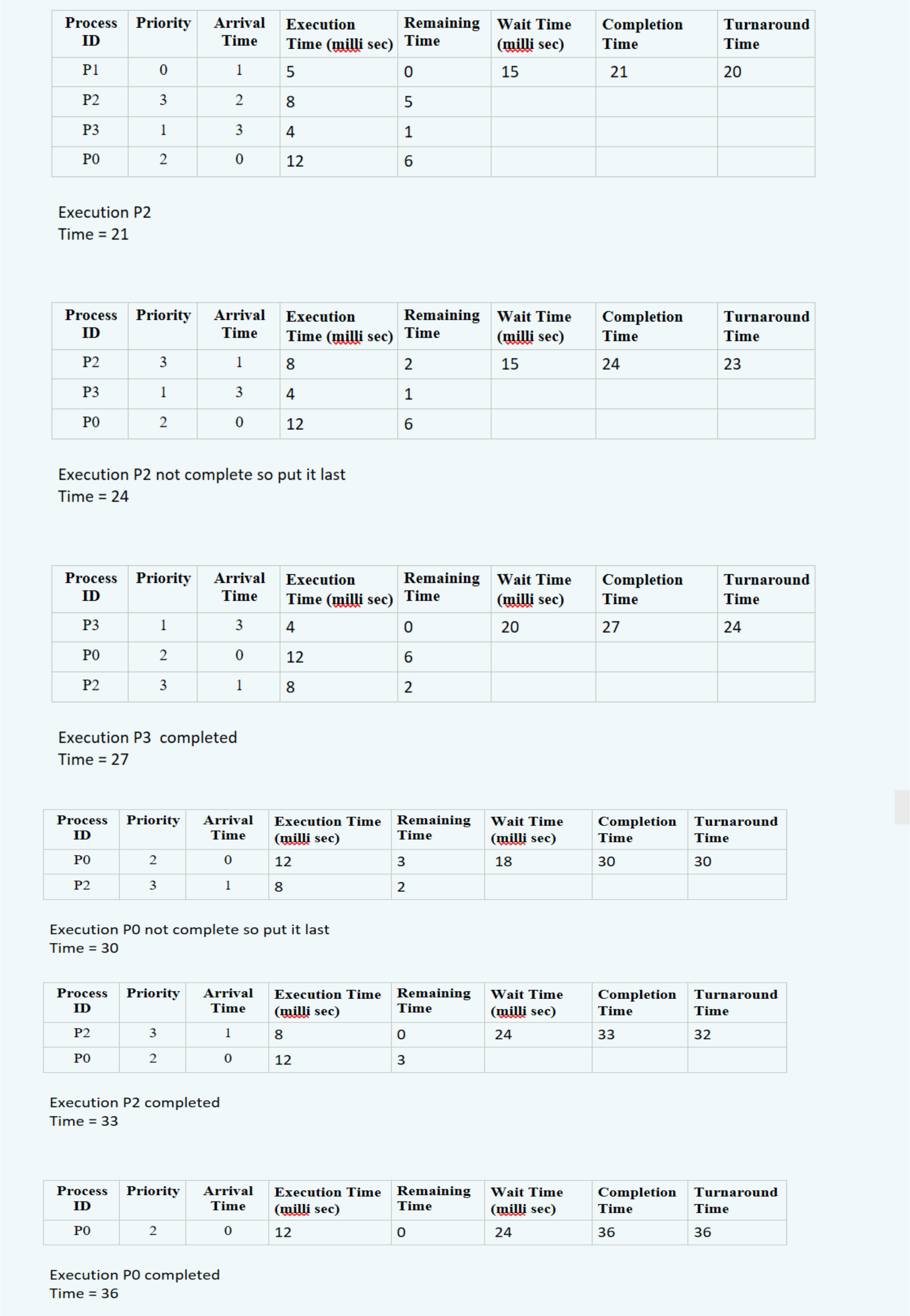
Time = 15

Execution time of P0 is over so P0 place at last

Process ID	Priority	Arrival Time	Execution Time (milli sec)	Remaining Time	Wait Time (milli sec)	Completion Time	Turnaround Time
P1	0	1	5	0	12	18	17
P2	3	2	8	5			
P3	1	3	4	1			
P0	2	3	12	6			

Executed of P1 so we can Dqueue it

Time = 18



Process ID	Priority	Arrival Time	Execution Time (milli sec)	Remaining Time	Wait Time (milli sec)	Completion Time	Turnaround Time
P1	0	1	5	0	15	21	20
P2	3	2	8	5			
P3	1	3	4	1			
P0	2	0	12	6			

Execution P2
Time = 21

Process ID	Priority	Arrival Time	Execution Time (milli sec)	Remaining Time	Wait Time (milli sec)	Completion Time	Turnaround Time
P2	3	1	8	2	15	24	23
P3	1	3	4	1			
P0	2	0	12	6			

Execution P2 not complete so put it last
Time = 24

Process ID	Priority	Arrival Time	Execution Time (milli sec)	Remaining Time	Wait Time (milli sec)	Completion Time	Turnaround Time
P3	1	3	4	0	20	27	24
P0	2	0	12	6			
P2	3	1	8	2			

Execution P3 completed
Time = 27

Process ID	Priority	Arrival Time	Execution Time (milli sec)	Remaining Time	Wait Time (milli sec)	Completion Time	Turnaround Time
P0	2	0	12	3	18	30	30
P2	3	1	8	2			

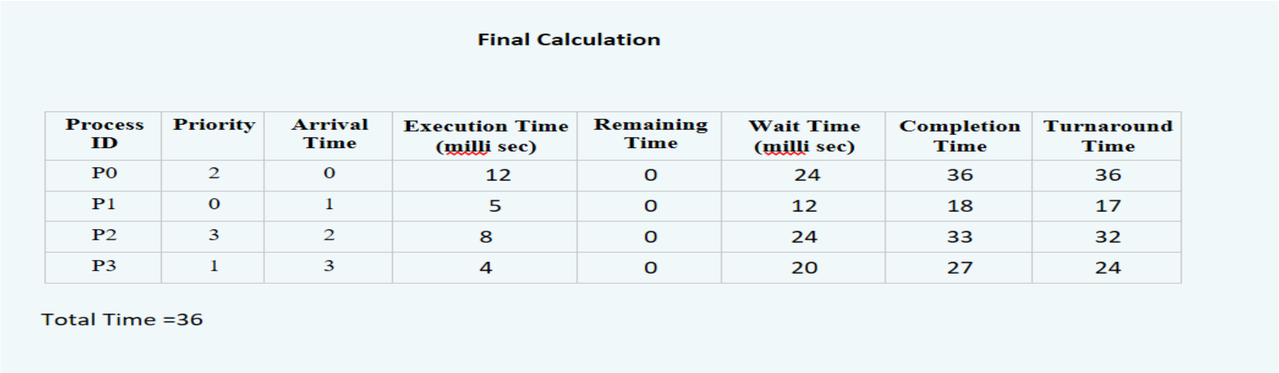
Execution P0 not complete so put it last
Time = 30

Process ID	Priority	Arrival Time	Execution Time (milli sec)	Remaining Time	Wait Time (milli sec)	Completion Time	Turnaround Time
P2	3	1	8	0	24	33	32
P0	2	0	12	3			

Execution P2 completed
Time = 33

Process ID	Priority	Arrival Time	Execution Time (milli sec)	Remaining Time	Wait Time (milli sec)	Completion Time	Turnaround Time
P0	2	0	12	0	24	36	36

Execution P0 completed
Time = 36



Final Calculation

Process ID	Priority	Arrival Time	Execution Time (milli sec)	Remaining Time	Wait Time (milli sec)	Completion Time	Turnaround Time
P0	2	0	12	0	24	36	36
P1	0	1	5	0	12	18	17
P2	3	2	8	0	24	33	32
P3	1	3	4	0	20	27	24

Total Time =36

The End