

Part I.a) : FCFS (First Come First Serve) Algorithm: (Unsure if I also had to explain but included it)

FCFS is a non-preemptive CPU scheduling algorithm that essentially executes processes in the order they arrive in the ready queue. So they follow the concept of First In First Out (FIFO).

The steps of FCFS are:

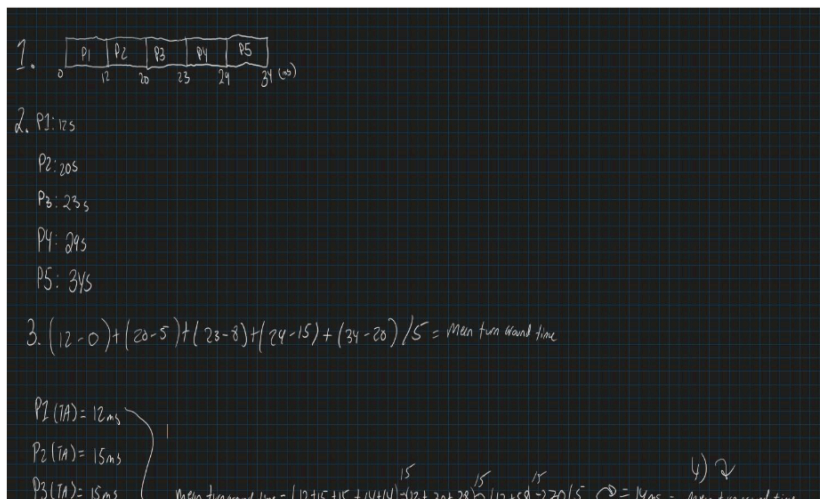
1. The processes are added to the ready queue in their order of arrival
2. The CPU scheduler (short-term scheduler) will then select the process at the head of the queue.
3. The selected process will execute until its completion or I/O request.
4. When completed or blocked, the next process in the ready queue will get the CPU.
5. This process will continue until all the processes are complete.

It is important to take into consideration those characteristics in FCFS. First, non-preemptive means that once a process gets the CPU, it will keep running until it terminates (so completion) or until it blocks/waits due to I/O (leaves CPU to wait for I/O). Also, every process gets CPU time. Since every process is served strictly in arrival order, it could lead to convoy effect. This occurs when a large process takes on the CPU for a large amount of time leaving a trail of smaller processes waiting for its completion (not efficient).

When a process requests for I/O, its state changes from running to waiting. Then the process is moved from the CPU to the I/O waiting queue. After that, the next process in ready queue gets the CPU. When the I/O completes, the process will return to the tail of ready queue and the state will change from waiting to ready.

Advantages of FCFS are that it's simple to implement and understand and that it has a predictable execution order. The disadvantages are the convoy effect and longer average turnaround times.

Gantt Chart + Calculations :



$P_4(T_4) = 14\text{ms}$
 $PSL(P_4) = 14\text{ms}$

20
 14

ii)

Round Robin is a preemptive CPU scheduling algorithm and it is made to give each process a fair slice of CPU time. It is one of the most used scheduling techniques in time-sharing systems. It enforces a strict time quantum/time slice for each process.

The steps of Round Robin are:

1. Processes enter the ready queue in their order of arrival, just like FCFS.
2. The CPU scheduler then chooses the process at the head of the ready queue.
3. The selected process then executes for at most one time quantum. If the process completes before the quantum expires, it will leave the system. If the process requests for I/O before the quantum ends, it will block and move to the i/o waiting queue. If the quantum ends and the process is not finished, it will be preempted and placed at the tail of the ready queue.
4. The CPU then chooses the next process at the head of the queue.
5. When a blocked process terminates its I/O, it returns to the tail of the ready queue.
6. All of this will happen over and over again until all processes are complete.

It is important to take into account a few aspects in Round Robin. First, a process can't monopolize the CPU due to the preemptive nature and its fairness. Once the time slice is over, the current process is preempted and the next process runs for its time slice. There are also more context switches because processes are preempted often. Finally, when a process requires I/O, it leaves the CPU and the next process runs while the I/O occurs. Once the I/O is complete, the process goes back to the tail of the ready queue.

