Tutorial-5 solution

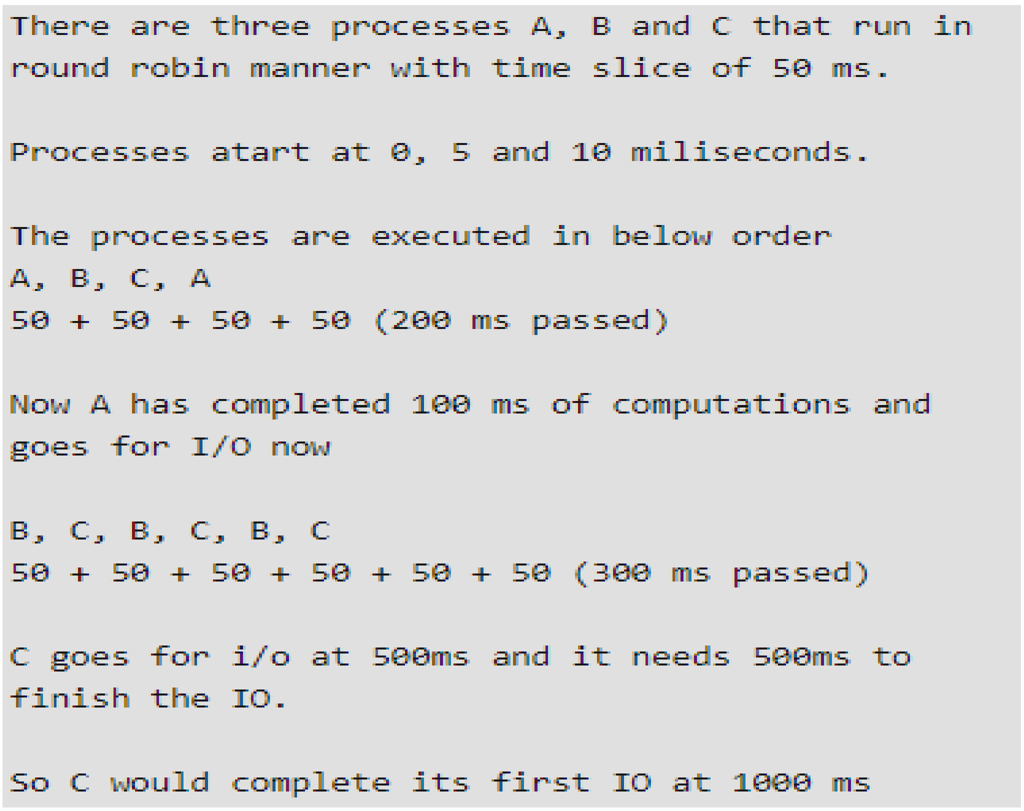
1. Three processes A, B and C each execute a loop of 100 iterations. In each iteration of the loop, a process performs a single computation that requires tc CPU milliseconds and then initiates a single I/O operation that lasts for tio milliseconds. It is assumed that the computer where the processes execute has enough I/O devices and the OS of the computer assigns different I/O devices to each process. Also, the scheduling overhead of the OS is negligible. The processes have the following characteristics:

PID	tc	tio
A	100 ms	500 ms
B	350 ms	500 ms
C	200 ms	500 ms

The processes A, B, and C are started at times 0, 5 and 10 milliseconds respectively, in a pure time-sharing system (round robin scheduling) that uses a time slice of 50 milliseconds. The time in milliseconds at which process C would complete its first I/O operation is ___________.

a. 500
b. 1000
c. 2000
d. 10000

Solution:



There are three processes A, B and C that run in round robin manner with time slice of 50 ms.

Processes atart at 0, 5 and 10 miliseconds.

The processes are executed in below order
A, B, C, A
50 + 50 + 50 + 50 (200 ms passed)

Now A has completed 100 ms of computations and goes for I/O now

B, C, B, C, B, C
50 + 50 + 50 + 50 + 50 + 50 (300 ms passed)

C goes for i/o at 500ms and it needs 500ms to finish the IO.

So C would complete its first IO at 1000 ms

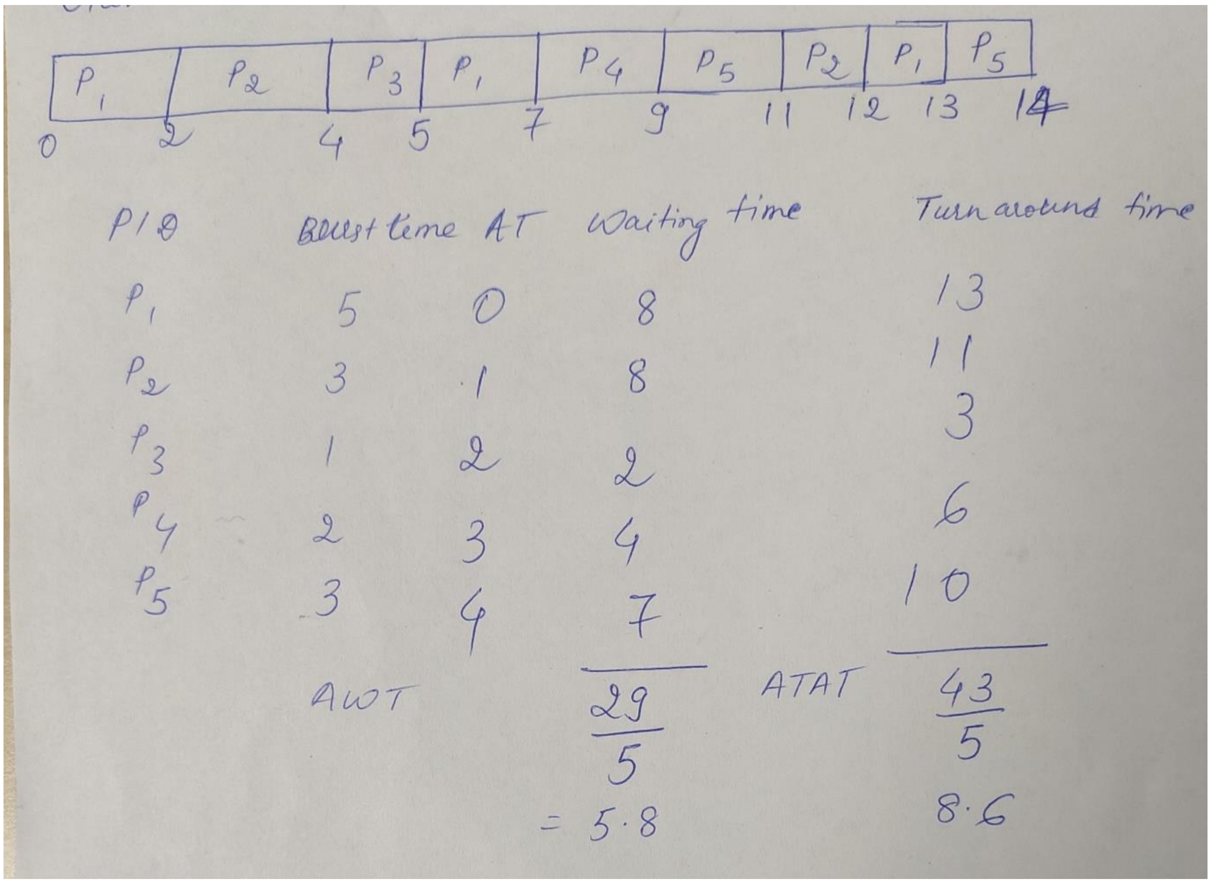
2. Consider the set of 5 processes whose arrival time and burst time are given below-

PID	Arrival time	Burst time
P1	0	5

P2	1	3
P3	2	1
P4	3	2
P5	4	3

If the CPU scheduling policy is Round Robin with time quantum = 2 unit, calculate the average waiting time and average turn-around time.

Solution:



P1	P2	P3	P1	P4	P5	P2	P1	P5

0 2 4 5 7 9 11 12 13 14

P/ID	Burst time	AT	Waiting time	Turn around time
P1	5	0	8	13
P2	3	1	8	11
P3	1	2	2	3
P4	2	3	4	6
P5	3	4	7	10

AWT $\frac{29}{5} = 5.8$

ATAT $\frac{43}{5}$ 8.6

3. Consider the 3 processes, P1, P2 and P3 with its arrival time and burst time shown in the table

PID	Arrival Time	Burst Time
P1	0	5
P2	1	7
P3	3	4

The completion order of the 3 processes under the policies FCFS and RR2 (round robin scheduling with CPU quantum of 2-time units) are __________.

Solution:

For FCFS, the order is clear.

In RR, time slot is of 2 units.

Processors are assigned in the following order

P1,P2,P1,P3,P2,P1,P3,P2,P2

At t = 2, P2 starts and P1 is sent to the ready queue and t = 3 P3 arrives so then the job P3 is queued in ready queue after P1, So at t = 4, again P1 is executed for first time at t = 6.

4. A uni-processor computer system only has two processes, both of which alternate 10ms CPU bursts with 90ms I/O bursts. Both the processes were created at nearly the same time. The I/O of both processes can proceed in parallel. Which of the following scheduling strategies will result in the least CPU utilization (over a long period of time) for this system?
   a. First come first served scheduling
   b. Shortest remaining time first scheduling
   c. Static priority scheduling with different priorities for the two processes
   d. Round robin scheduling with a time quantum of 5 ms

   Solution:
   We are given that the time slice is 5ms. Consider process P and Q. Say P utilizes 5ms of CPU and then Q utilizes 5ms of CPU. Hence after 15ms P starts with I/O And after 20ms Q also starts with I/O. Since I/O can be done in parallel, P finishes I\O at 105th ms (15 + 90) and Q finishes its I\O at 110th ms (20 + 90). Therefore, we can see that CPU remains idle from 20th to 105th ms.

   That is when Round Robin scheduling is used,
   Idle time of CPU = 85ms
   CPU Utilization = 20/105 = 19.05%
   When First Come First Served scheduling scheduling or Shortest Remaining Time First is used
   Say P utilizes 10ms of CPU and then starts its I/O. At 11th ms Q starts processing. Q utilizes 10ms of CPU.
   P completes its I/O at 100ms (10 + 90)
   Q completes its I/O at 110ms (20 + 90)
   At 101th ms P again utilizes CPU. Hence,
   Idle time of CPU = 80ms
   CPU Utilization = 20/100 = 20%

   Since only two processes are involved and I\O time is much more than CPU time, “Static priority scheduling with different priorities” for the two processes reduces to FCFS or Shortest remaining time first.

   Therefore, Round robin will result in least CPU utilization.

5. A system uses the following preemptive priority scheduling algorithm (process with larger priority numbers have higher priority). Processes enter the system with a priority of 1. While waiting on the ready queue, a process’s priority changes at rate α. While running, a process’s priority changes at the rate β.
   a) What is the algorithm that results from $\beta > \alpha > 1$?
   b) What is the algorithm that results from $\alpha < \beta < 1$?

   Answer:

a) While in the ready state, a process's priority increases. Thus, the process that has been in the ready state the longest will have the highest priority. The process that is running has its priority increased at the rate faster than any process in the ready state. So, no process in the ready state will ever preempt a running process. Running to termination, the process that has been in the ready state the longest is First-Come-First-Served.

b) As soon as a process enters the system, its priority starts decreasing. A relatively new process will have the highest priority and be immediately made runnable. When a running process terminates, the process that has been in the ready queue the shortest will have the highest priority. The algorithm is Last-In-First-Out.

6. On a system using round-robin scheduling, let s represent the time needed to perform a process switch, q represent the round- robin time quantum, and r represent the average time a process runs before blocking on I/O. Give a formula for CPU efficiency given the following.
   a) $q = \infty$
   b) $q = r$
   c) $s < q < r$
   d) $s = q < r$

   Solution:

   a) Processes will run until they block. For each cycle, s units of overhead will be needed to accomplish r units of useful work. CPU efficiency is $r/ (r + s)$.

   b) Since processes will still run until they block, the answer is the same as for part (a).
   c) The number of switches required will be r/q, making the time wasted on switches sr/q. CPU efficiency is $r/(r + sr/q) = q/(q +s)$.

   d) Same answer as above except with $q = s$, the equation evaluates to ½.