**OSG202 - Lab 4**

**Process scheduling**

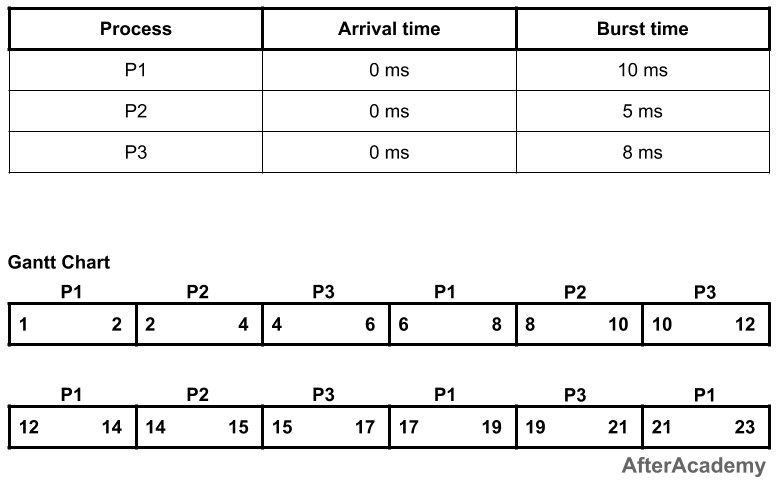
|  |  |
| --- | --- |
| Kiểm tra gcc đã có chưa:  **which gcc**  hiển thị:  **/usr/bin/gcc**  là đã có.  Cài đặt gcc:  **sudo apt -get install gcc** | Biên dịch test1.c  **gcc -c test1.c –o test1.o**  Chuyển thành file \*.exe:  **gcc -o test1 test1.o**  Thực thi chương trình  **./test1** |

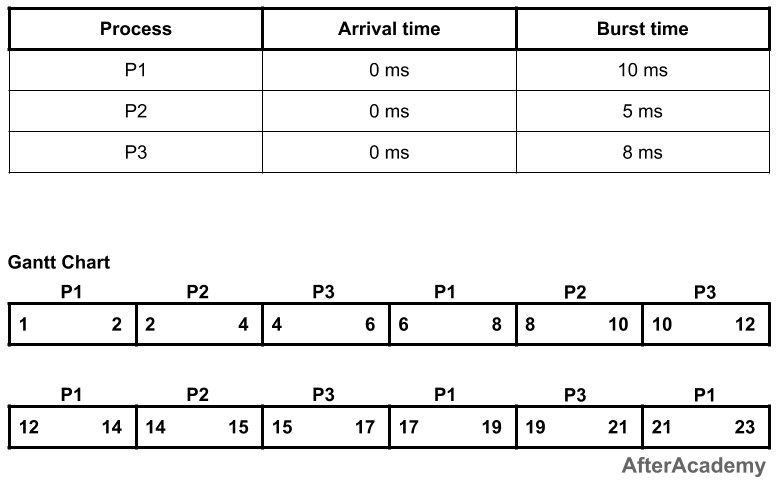
**Round-Robin scheduling**

In this approach of CPU scheduling, we have a fixed time quantum and the CPU will be allocated to a process for that amount of time only at a time. For example, if we are having three process P1, P2, and P3, and our time quantum is 2ms, then P1 will be given 2ms for its execution, then P2 will be given 2ms, then P3 will be given 2ms. After one cycle, again P1 will be given 2ms, then P2 will be given 2ms and so on until the processes complete its execution.

It is generally used in the time-sharing environments and there will be no starvation in case of the round-robin.

**Example:**





In the above example, every process will be given 2ms in one turn because we have taken the time quantum to be 2ms. So process P1 will be executed for 2ms, then process P2 will be executed for 2ms, then P3 will be executed for 2 ms. Again process P1 will be executed for 2ms, then P2, and so on. The waiting time and turnaround time of the processes will be:

---------------------------------------------

| Process | Waiting Time | Turnaround Time |

---------------------------------------------

| P1 | 13ms | 23ms |

| P2 | 10ms | 15ms |

| P3 | 13ms | 21ms |

---------------------------------------------

Total waiting time: (13 + 10 + 13) = 36ms

Average waiting time: (36/3) = 12ms

Total turnaround time: (23 + 15 + 21) = 59ms

Average turnaround time: (59/3) = 19.66ms

**Lab 4 requirements:**

Consider the following processes:

char name[50][10] = {"A","B","C","D","E"}; // process names

int btime[] = {9,7,8,5,3}; // process burst times

int n = 5;

Write program in gcc language to give the following output:

