WEEK 5 OS LAB

Yash Gupta
1BM21CS251

12-07-2023

Q: 2. Simulate Rate Monotonic Scheduling for the following and show the order of execution of processes in CPU timeline:

Process	Execution Time	Period
P_1	3	20
P_2	2	5
P_3	2	10

```
#include <stdio.h>
struct process
{
   int B_time;
   int period;
   int pid;
   int count;
};
typedef struct process procs;
int lcm(int n1, int n2, int n3)
{
   int max = n1;
   if (n2 > max)
   {
```

```
        max = n2;
    }
    if (n3 > max)
    {
        max = n3;
    }
    for (int i = max;; i++)
    {
        if (i % n1 == 0 && i % n2 == 0 && i % n3 == 0)
        {
            return i;
        }
    }
}
int main()
{
    int n;
    printf("enter the number of the processes\n");
    scanf("%d", &n);
    procs processes[n];
    for (int i = 0; i < n; i++)
    {
        printf("enter the execution time for process:%d\n", i + 1);
        scanf("%d", &processes[i].B_time);
        printf("enter time period for process:%d\n", i + 1);
        scanf("%d", &processes[i].period);
        processes[i].pid = i + 1;
    }
    procs temp;
```

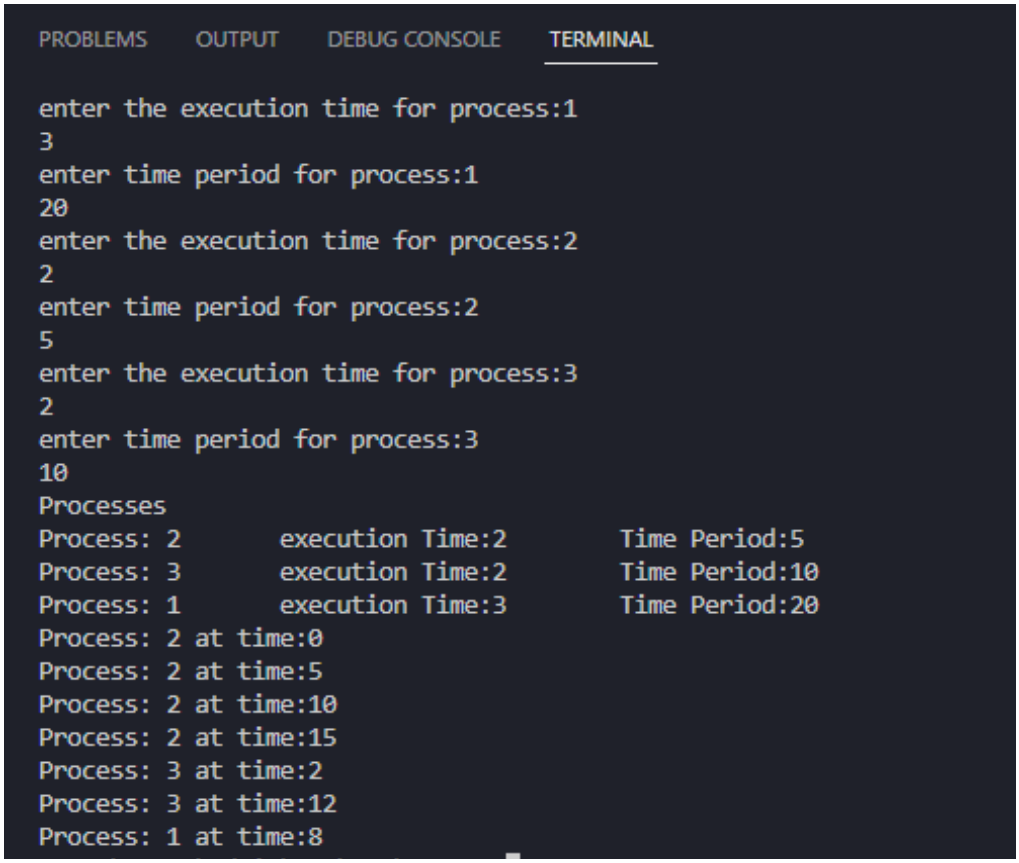
```
    for (int i = 0; i < n - 1; i++)
    {
        for (int j = 0; j < n - i - 1; j++)
        {
            if (processes[j + 1].period < processes[j].period)
            {
                temp = processes[j];
                processes[j] = processes[j + 1];
                processes[j + 1] = temp;
            }
        }
    }
    printf("Processes\n");
    int Ftime = lcm(processes[0].period, processes[1].period,
processes[2].period);
    for (int i = 0; i < n; i++)
    {
        printf("Process: %d\t execution Time:%d\t Time Period:%d\n",
processes[i].pid, processes[i].B_time, processes[i].period);
        processes[i].count = Ftime / processes[i].period;
    }
    for (int i = 0; i < n; i++)
    {
        int j = 0;
        if (i == 0)
        {
            int cnt = processes[i].count;
            while (j < cnt)
            {
```

```
            printf("Process: %d at time:%d\n", processes[i].pid,
(processes[i].period) * j);
            j++;
        }
    }
    if (i == 1)
    {
        int cnt = processes[i].count;
        while (j < cnt)
        {

            printf("Process: %d at time:%d\n", processes[i].pid,
((processes[i].period) * j)+processes[i-1].B_time);
            j++;
        }
    }
    if(i==2)
    {
        int cnt = processes[i].count;
        while (j < cnt)
        {
            int time = 8;
            printf("Process: %d at time:%d\n", processes[i].pid, time);
            j++;
        }
    }
  }
  return 0;
```

}

OUTPUT:


PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
enter the execution time for process:1
3
enter time period for process:1
20
enter the execution time for process:2
2
enter time period for process:2
5
enter the execution time for process:3
2
enter time period for process:3
10
Processes
Process: 2 execution Time:2 Time Period:5
Process: 3 execution Time:2 Time Period:10
Process: 1 execution Time:3 Time Period:20
Process: 2 at time:0
Process: 2 at time:5
Process: 2 at time:10
Process: 2 at time:15
Process: 3 at time:2
Process: 3 at time:12
Process: 1 at time:8