**WEEK 5 OS LAB**

**Yapara karthikeya**

**1BM21CS249**

**12-07-2023**

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



#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++;

}

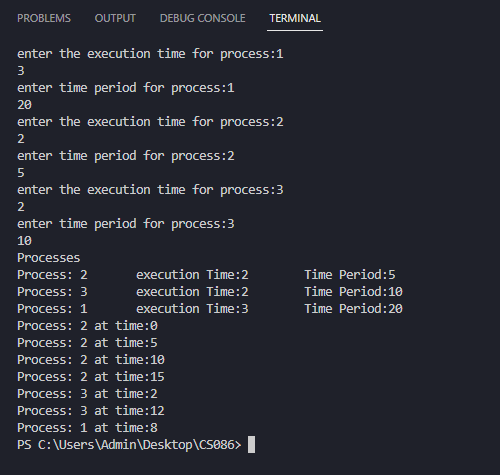
}

}

return 0;

}

**OUTPUT:**

****