Lab 4: Write a C program to simulate Real-Time CPU Scheduling algorithms: b) Earliest-deadline First

19/7/23.

Q4b) Write a C program to simulate Real-Time CPU scheduling algorithms:

b) Earliest-deadline First.

```
#include <stdio.h>

#define MAX_TASKS 100

typedef struct
{
    int task_id;
    int arrival_time;
    int execution_time;
    int deadline;
    int is_completed;
} Task;

float calculate_cpu_utilization(Task process[], int n)
{
    float total_utilization = 0.0;

    for (int i = 0; i < n; i++)
    {
        float ~~int~~ i task_utilization = (float) process[i].
        execution_time / process[i].deadline;
        total_utilization += task_utilization;
    }
    float cpu_utilization = total_utilization * 100;
    return cpu_utilization;
}
```

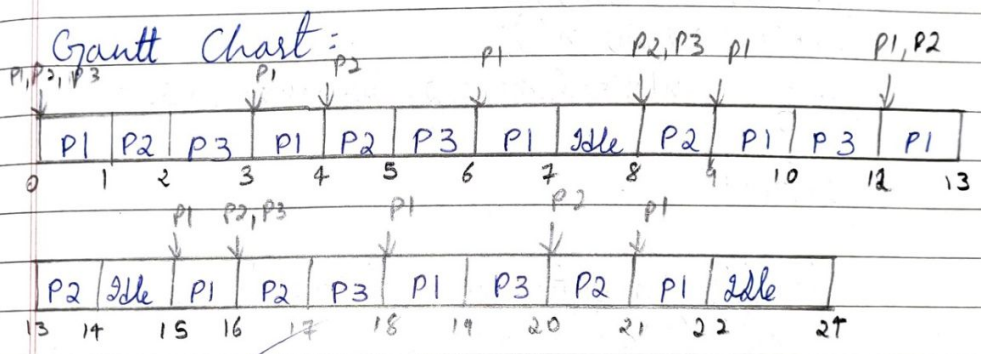
```
int main()
{
  int n, i;

  printf("Enter the number of processes: ");
  scanf("%d", &n);

  Task process[MAX_TASKS];
  // Read task details
  for (i=0; i<n; i++)
  {
    printf("Task %d\n", i+1);
    printf("Enter execution time: ");
    scanf("%d", &process[i].execution_time);
    printf("Enter deadline: ");
    scanf("%d", &process[i].deadline);

    process[i].task_id = i+1;
    process[i].is_completed = 0;

  float cpu_utilization = calculate_cpu_utilization
                          (process, n);

  printf("CPU Utilization: %.2f%%\n",
         cpu_utilization);

  return 0;
}
```

store
67

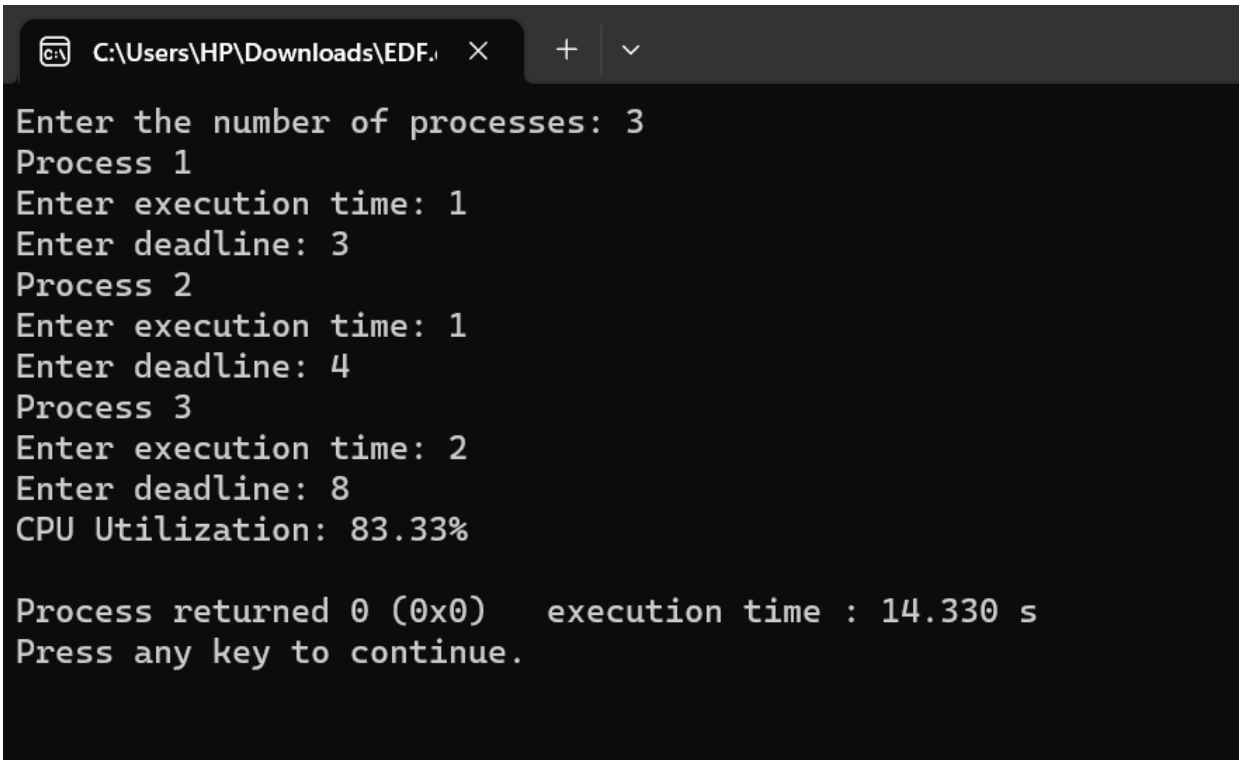
Output:

```
Enter the number of processes : 3
Process 1
Enter execution time : 1
Enter deadline : 3
Process 2
Enter execution time : 1
Enter deadline : 4
Process 3
Enter execution time: 2
Enter deadline : 8
CPU Utilization: 83.33%
```

Gantt Chart:



OUTPUT :


C:\Users\HP\Downloads\EDF.
Enter the number of processes: 3
Process 1
Enter execution time: 1
Enter deadline: 3
Process 2
Enter execution time: 1
Enter deadline: 4
Process 3
Enter execution time: 2
Enter deadline: 8
CPU Utilization: 83.33%
Process returned 0 (0x0) execution time : 14.330 s
Press any key to continue.

5. Write a C program to simulate producer-consumer problem using semaphores.

19/7/23

Q5) Write a C program to simulate producer-consumer problem using semaphores.

```
#include <stdio.h>
#include <stdlib.h>

int mutex = 1, full = 0, empty = 3, x = 0;

int main()
{
  int n;
  void producer();
  void consumer();
  int wait(int);
  int signal(int);
  printf("MENU\n");
  printf("\n1. Producer\n2. Consumer\n3. Exit");

while(1)
{
 printf("\n Enter your choice:");
 scanf("%d",&n);
switch(n)
{
 case 1: if(mutex==1 && empty!=0)
           producer();
         else
           printf("Buffer is full!!");
         break;

case 2: if(mutex==1 && full!=0)
          cosumer();
```

```
        else
            printf("Buffer is empty!");
        break;
    case 3:
        exit(0);
        break;
    }
}
return 0;
}

int wait(int s)
{
  return (--s);
}

int signal(int s)
{
  return (++s);
}

void producer()
{
  mutex = wait(mutex);
  full = signal(full);
  empty = wait(empty);
  x++;
  printf("\n Producer produces the item %d", x);
  mutex = signal(mutex);
}
```

```
void consumer()
{
    mutex = wait(mutex);
    full = wait(full);
    empty = signal(empty);
    printf("\n Consumer consumes item %d", x);
    x--;
    mutex = signal(mutex);
}
```

Output:

MENU
1. Producer
2. Consumer
3. Exit

Enter your choice: 2
Buffer is Empty!

Enter your choice: 1
Producer produces item 1

Enter your choice: 1
Producer produces item 2

Enter your choice: 2
Consumer consumes item 2

Enter your choice: 1
Producer produces item 2

Enter your choice : 2
Consumer consumes item 2

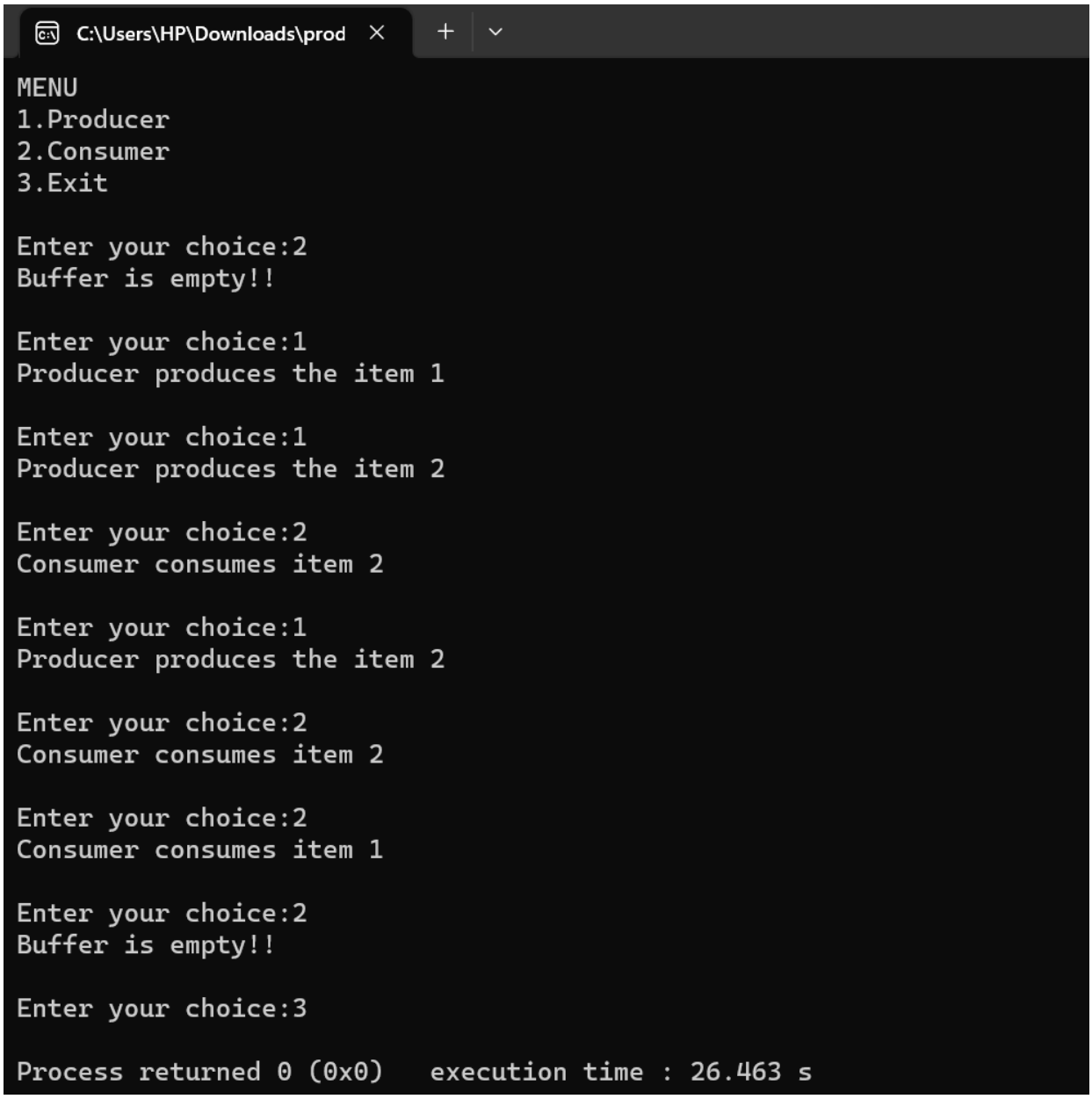
Enter your choice : 2
Consumer consumes item 1

Enter your choice : 2
Buffer is empty !

Enter your choice : 3

19/7/23

OUTPUT :


C:\Users\HP\Downloads\prod
MENU
1.Producer
2.Consumer
3.Exit
Enter your choice:2
Buffer is empty!!
Enter your choice:1
Producer produces the item 1
Enter your choice:1
Producer produces the item 2
Enter your choice:2
Consumer consumes item 2
Enter your choice:1
Producer produces the item 2
Enter your choice:2
Consumer consumes item 2
Enter your choice:2
Consumer consumes item 1
Enter your choice:2
Buffer is empty!!
Enter your choice:3
Process returned 0 (0x0) execution time : 26.463 s