# Mawlana Bhashani Science and Technology University

# Lab-Report

Lab Report No: 10

Lab Report Name: Implementation of Round Robin Scheduling Algorithm.

Course code: ICT-3110

Course title: Operating System Lab

Date of Performance: 7/9/2020

Date of Submission:

## Submitted by

Name: Afra Ibnat Tethye

ID: IT-18055

3rd year 1st semester

Session: 2017-18

Dept. of ICT

## Submitted To

Nazrul Islam

Assistant Professor

Dept. of ICT,

MBSTU.

**Experiment Name:** Implementation of Round Robin Scheduling Algorithm.

## Question-01:

### What is Round Robin Scheduling algorithm?

## Answer:

Rouond Robin is the preemptive process scheduling algorithm. Each process is provided a fix time to execute, it is called a quantum. Once a process is executed for a given time period , it is preempted and other process executes for a given period time period.

## Question-02:

### How to implemented in C?

## Answer:

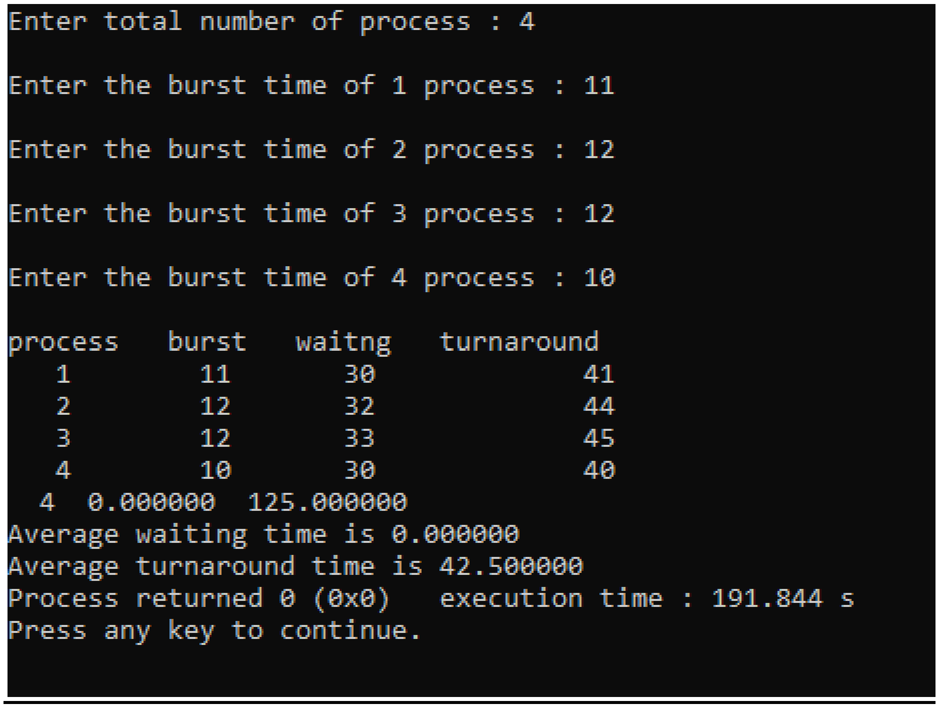
The code is given bellow:

```
#include<stdio.h>
int main() {
  int burst [100],waiting [100],turnaround [100],b[100];
  int i,n,time,count=0;
  float total=0,totalTT=0,avgwt,avgtt,totalwt=0;

   printf("Enter total number of process : ");
      scanf("%d",&n);
     for(i=0; i<n; i++)
  {
```

```
printf("\nEnter the burst time of %d process : ",i+1);

scanf("%d",&burst [i]);

b[i] = burst [i];

   }    i=0;

   for(time=0; count!=n; time++)

   {

      while(burst [i] == 0)

      {           i=(i+1)%n;

      }       burst [i]--;       if(burst [i]==0)

      {

         turnaround [i] = time+1;            count++;

      }       i = (i+1)%n;

   }

   printf("\nprocess   burst   waitng   turnaround   ");    for(i=0; i<n; i++)

   {

waiting [i] = turnaround [i] - b[i];

      printf("\n   %d \t    %d \t     %d \t    %d",i+1,b[i],waiting [i],turnaround [i]);

       total = total + waiting [i];

       totalTT = totalTT + turnaround [i];

   }

   printf("\n  %d  %f  %f",n,totalwt,total);

   avgwt = totalwt / n;

   avgtt = totalTT / n;

   printf("\nAverage waiting time is %f",avgwt);
```

```
    printf("\nAverage turnaround time is %f ",avgtt);

    return 0;

}
```

## <u>Output:</u>



```
Enter total number of process : 4

Enter the burst time of 1 process : 11

Enter the burst time of 2 process : 12

Enter the burst time of 3 process : 12

Enter the burst time of 4 process : 10

process   burst   waitng   turnaround
   1        11       30             41
   2        12       32             44
   3        12       33             45
   4        10       30             40
  4  0.000000  125.000000
Average waiting time is 0.000000
Average turnaround time is 42.500000
Process returned 0 (0x0)   execution time : 191.844 s
Press any key to continue.
```