MAWLANA BHASHANI SCIENCE AND TECHNOLOGY UNIVERSITY

Santosh,Tangail-1902

LAB REPORT

Lab Report No	:	10
Lab Report name	:	Implementation of Round Robin Scheduling algorithm.
Course Title	:	Operating System Lab
Course Code	:	ICT-3110
Date of Performance	:	20-08-2020
Date of Submission	:	28-08-2020

Submitted by,

Student Name	:	Tanvir Ahmed
Student ID	:	IT-18043
Session	:	2017-18

3rd Year 1st semester

Dept. of ICT

Submitted to,

Nazrul Islam

Assistant Professor

Dept. of ICT,

MBSTU.

Lab Report No. **10**

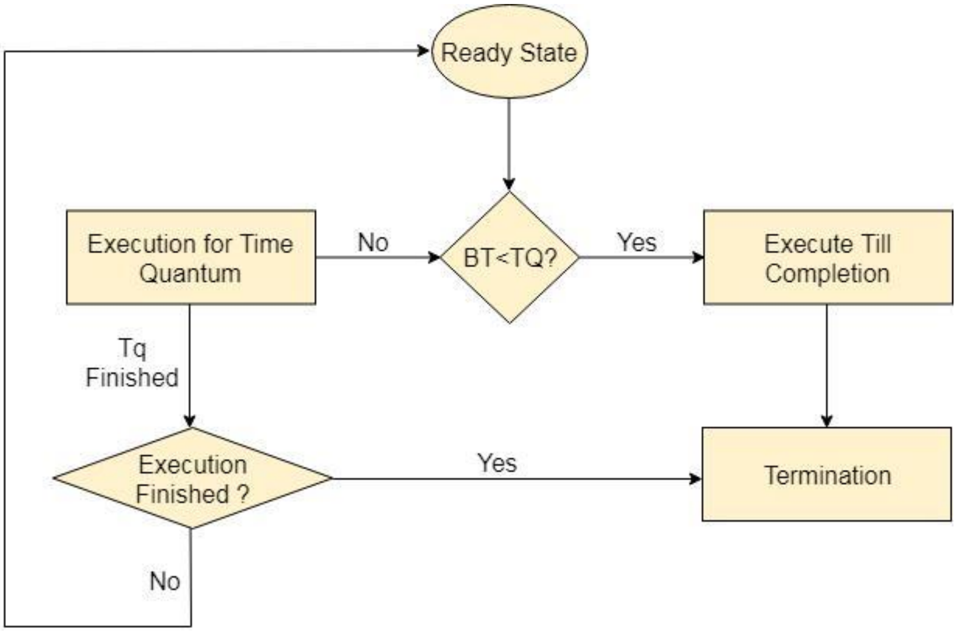
Lab Report Name: **Implementation of Round Robin Scheduling algorithm .**

Objectives:

i. What is Round Robin Scheduling algorithm.
ii. How to implementation

Theory : Round robin is the most widely used process scheduling algorithm .The basic strategy for round robin scheduling is that if there are n process,each of the process will receive 1/n CPU Execution Time.Each process is allotted a time quanta, for which its is executed.The incoming processes are kept in a ready list while another one is executing.If the time quanta allotted for a process is over,then that process is moved to ready and the next process in the ready list is executed for the allotted time quanta.

The Complete Example Implementation Source Code in C of Round Robin Algorithm to schedule N Processes and to calculate the Execution,wait time and turn around time is given below.



Characteristics of Round-Robin Scheduling

Here are the important characteristics of Round-Robin Scheduling:

1. Round robin is a pre-emptive algorithm.
2. The CPU is shifted to the next process after fixed interval time, which is called time quantum/time slice.
3. The process that is preempted is added to the end of the queue.
4. Round robin is a hybrid model which is clock-driven
5. Time slice should be minimum, which is assigned for a specific task that needs to be processed. However, it may differ OS to OS.
6. It is a real time algorithm which responds to the event within a specific time limit.
7. Round robin is one of the oldest, fairest, and easiest algorithm.
8. Widely used scheduling method in traditional OS.

Advantages

1.It can be actually implementable in the system because it is not depending on the burst time.

2.It doesn't suffer from the problem of starvation or convoy effect.

3.All the jobs get a fare allocation of CPU.

Disadvantages

1.The higher the time quantum, the higher the response time in the system.

2.The lower the time quantum, the higher the context switching overhead in the system.

3.Deciding a perfect time quantum is really a very difficult task in the system.

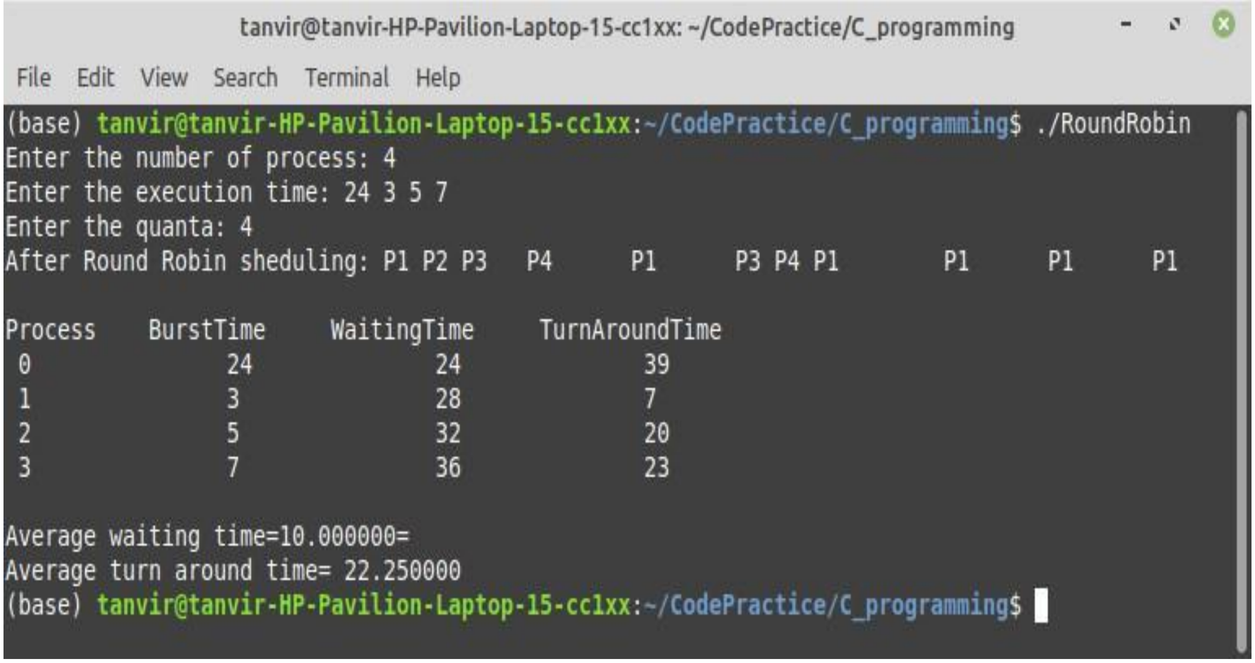
Corresponding Code:

```
#include<stdio.h>

int main()

{

   int n,i,k,x=0,s=0,r=0,q=0,a[30],e[30],t[30];

   float m,p=0;
```

```
printf("Enter the number of process: ");

scanf("%d",&n);

printf("Enter the execution time: ");

for(i=0; i<n; i++)

{

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

   e[i]=a[i];

}

printf("Enter the quanta: ");

scanf("%d",&q);

printf("After Round Robin sheduling: ");

for(i=0; i<n; i++)

{

   if(x<a[i])

   {

      x=a[i];

   }

}

k=x/q;

while(s<=k)

{

   for(i=0; i<n; i++)

   {

      if(a[i]>0)

      {

         if(a[i]>q)
      {
        r=r+q;
        a[i]=a[i]-q;
        printf("P%d\t",i+1);
      }else
      {
        r=r+a[i];
        a[i]=a[i]-q;
        printf("P%d ",i+1);
        t[i]=r;
      }
    }
  }
  s++;
}
printf("\n\nProcess     BurstTime     WaitingTime     TurnAroundTime\n");
for(i=0; i<n; i++)
{
  printf(" %d \t\t %d\t\t %d\t\t %d\t\t \n",i,e[i],x,t[i]);
  x=x+q;
}
m=x/n;
printf("\nAverage waiting time=%f= ",m);
printf("\nAverage turn around time= ");
for(i=0; i<n; i++)
  p=p+t[i];
```

```
    p=p/n;

    printf("%f",p);

    printf("\n");

    return 0;

}
```

Output:


tanvir@tanvir-HP-Pavilion-Laptop-15-cc1xx: ~/CodePractice/C_programming
File
Edit
View
Search
Terminal
Help

```
(base) tanvir@tanvir-HP-Pavilion-Laptop-15-cc1xx:~/CodePractice/C_programming$ ./RoundRobin
Enter the number of process: 4
Enter the execution time: 24 3 5 7
Enter the quanta: 4
After Round Robin sheduling: P1 P2 P3   P4      P1      P3 P4 P1       P1      P1      P1

Process    BurstTime     WaitingTime     TurnAroundTime
 0               24             24              39
 1               3              28              7
 2               5              32              20
 3               7              36              23

Average waiting time=10.000000=
Average turn around time= 22.250000
(base) tanvir@tanvir-HP-Pavilion-Laptop-15-cc1xx:~/CodePractice/C_programming$ 
```

Conclusion : Round robin is one of the oldest, fairest, and easiest algorithms and widely used scheduling methods in traditional OS. The biggest advantage of the round-robin scheduling method is that If you know the total number of processes on the run queue, then you can also assume the worst-case response time for the same process.

Here we got 4 processes of burst time 24,3,5,7 where the quanta is 4 . So the cpu will execute for the time period of 4 for every process.When any process is completely executed then that will be not considered. Then other will be executed circularly .That’s the way the round robin algorithm works

Round-robin scheduling doesn't give special priority to more important tasks. Lower time quantum results in higher the context switching overhead in the system.