

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 :

Date of Submission : 29/9/2020

Submitted by,

Name : Farhana Afrin Shikha

ID :IT-18038

Session : 2017-18

3rd Year 1stsemester

Dept. of ICT

Submitted to,

Nazrul Islam

Assistant Professor

Dept. of ICT,

MBSTU.

**Lab Report 10 : Implementation of Round Robin Scheduling algorithm .**

**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.

**Advantages:**

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

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

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

**Disadvantage:**

1.This method spends more time on context switching.

2.Its performance heavily depends on yime quantum.

3.Priorities can’t be set for the processes.

Characteristics of Round-Robin Scheduling Here are the important characteristics of Round-Robin Scheduling:

1. Round robin is a pre-emptive algorithm.

2. It is a real time algorithm which responds to the event within a specific time limit.

3. Round robin is a hybrid model which is clock-driven

4. Round robin is one of the oldest, fairest, and easiest algorithm.

5. The CPU is shifted to the next process after fixed interval time, which is called time quantum/time slice.

6. Widely used scheduling method in traditional OS.

**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;

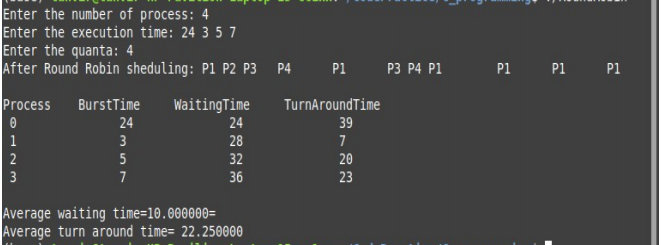
printf("%f",p);

printf("\n");

return 0;

}

**Output:**



**Conclusion:**Round Robin (RR) scheduling algorithm is the widely used scheduling algorithm in multitasking. It ensures fairness and starvation free execution of processes. Choosing the time quantum in RR algorithm is very crucial as small time slice results in large number of context switches and large time quantum increases the response time. To overcome these problems of RR scheduling, instead of static time slice dynamic time slice can be used to get optimal performance. The objective of this paper is to modify RR algorithm by adjusting time slices of different rounds depending on the remaining CPU bursts of currently running processes and considering their waiting times until that round in respect of the other processes’ waiting times. Experimental analysis reveals that the proposed algorithm produces better average turnaround time, average waiting time and fewer number of context switches than existing.RR scheduling algorithm is not so easy.It is harder than any other CPU scheduling.