Mawlana Bhashani Science and Technology University

Lab-Report

Report No : 10
Experiment name : Implementation of Round Robin Scheduling Algorithm
Course code : ICT-3110
Course title : Operating System Lab.
Date of Performance :
Date of Submission :

Submitted by

Name: Iqbal Hossen
ID: IT-18041
3rd year 1st semester
Session: 2017-18
Dept. of ICT
MBSTU.

Submitted To

Nazrul Islam
Assistant Professor
Dept. of ICT
MBSTU.

i) What is Round Robin Scheduling Algorithm?

Round Robin scheduling :

Let’s take one example to understand it.
Time Quantum = 2

Process	Arrival time	Burst time
P1	0	4
P2	1	5
P3	2	2
P4	3	1
P5	4	6
P6	5	3

Hence the GANTT chart will be following:

P1	P2	P3	P1	P4	P5	P2	P6	P5	P2	P6	P5

0 2 4 6 8 9 11 13 15 17 18 19 21

Process	A.T	B.T	W.T=(s.t-a.t) + (s.t-l.c.t)	T.A.T=B.T+W.T	R.T= s.t-a.t
P1	0	4	4	8	0
P2	1	5	12	17	1
P3	2	2	2	4	2
P4	3	1	5	6	5

P5	4	6	11	17	5
P6	6	3	10	13	7

Average waiting time = $\frac{4+12+2+5+11+10}{6}$ = 7.33 ms

Average turn around time = $\frac{8+17+4+6+17+13}{6}$ = 10.83 ms

ii) implementation of Round Robin algorithm in C

The implementation of Round Robin scheduling algorithm in C is given below:

Code:

```
//implementation of Round Robin scheduling algorithm
#include <iostream>
#include <vector>
using namespace std;
int main()
{
   int i,n,time,remain,temps=0,time_quantum;
   int wt=0,tat=0;
   cout<<"Enter the total number of process:";
   cin>>n;
   remain=n;

   vector<int>at(n);
   vector<int>bt(n);
   vector<int>rt(n);

   for(i=0; i<n; i++)
   {
      cout<<"Enter the Arrival time & Burst time of the processes:"<<i+1<<" ";
      cin>>at[i]>>bt[i];
      rt[i]=bt[i];
   }
   cout<<"Enter the value of time QUANTUM:"<<endl;
   cin>>time_quantum;
   cout<<"Process\tA.T\tB.T\tW.T\tT.A.T\n";
   for(time=0,i=0; remain!=0;)
   {
      if(rt[i]<=time_quantum && rt[i]>0)
```

```
        {
            time += rt[i];
            rt[i]=0;
            temps=1;
        }

        else if(rt[i]>0)
        {
            rt[i] -= time_quantum;
            time += time_quantum;
        }

        if(rt[i]==0 && temps==1)
        {
            remain--;
            cout<<i+1<<"\t"<<at[i]<<"\t"<<bt[i]<<"\t"<<time-at[i]-
bt[i]<<"\t"<<time-at[i];
            cout<<endl;
            wt += time-at[i]-bt[i];
            tat += time-at[i];
            temps=0;
        }

        if(i == n-1)
            i=0;
        else if(at[i+1] <= time)
            i++;
        else
            i=0;
    }

    cout<<"Average waiting time "<<wt*1.0/n<<endl;
    cout<<"Average turn around time "<<tat*1.0/n<<endl;;

    return 0;
}
```

Output:

```
Enter the total number of process:4
Enter the Arrival time & Burst time of the processes:1 1 4
Enter the Arrival time & Burst time of the processes:2 2 3
Enter the Arrival time & Burst time of the processes:3 3 5
Enter the Arrival time & Burst time of the processes:4 4 7
Enter the value of time QUANTUM:
2
Process A.T     B.T     W.T     T.A.T
1       1       4       5       9
2       2       3       6       9
3       3       5       8       13
4       4       7       8       15
Average waiting time 6.75
Average turn around time 11.5

Process returned 0 (0x0)   execution time : 49.919 s
```