

Rajneesh Pandey - 106119100

Question

Program for Round Robin scheduling |

Round Robin is a CPU scheduling algorithm where each process is assigned a fixed time slot in a cyclic way.

- It is simple, easy to implement, and starvation-free as all processes get fair share of CPU.
- One of the most commonly used technique in CPU scheduling as a core.
- It is preemptive as processes are assigned CPU only for a fixed slice of time at most.
- The disadvantage of it is more overhead of context switching.

Program and Input/Output

```

1 //106119100 Rajneesh Pandey
2 #include <iostream>
3 using namespace std;
4
5 void findWaitingTime(int processes[], int n, int bt[], int wt[], int quantum){
6     int rem_bt[n];
7     for (int i = 0; i < n; i++) rem_bt[i] = bt[i];
8     int t = 0;
9     while (1){ bool done = true;
10        for (int i = 0; i < n; i++){
11            if (rem_bt[i] > 0){done = false;
12                if (rem_bt[i] > quantum){ t += quantum;
13                    rem_bt[i] -= quantum; }
14                else{
15                    t = t + rem_bt[i]; wt[i] = t - bt[i]; rem_bt[i] = 0;}} }
16        if (done == true) break;}}
17
18 void findTurnAroundTime(int processes[], int n, int bt[], int wt[], int tat[]){
19     for (int i = 0; i < n; i++)
20         tat[i] = bt[i] + wt[i];
21
22 void findavgTime(int processes[], int n, int bt[], int quantum){
23     int wt[n], tat[n], total_wt = 0, total_tat = 0;
24     findWaitingTime(processes, n, bt, wt, quantum);
25     findTurnAroundTime(processes, n, bt, wt, tat);
26     cout << "Processes " << " Burst time " << " Waiting time " << " Turn around time\n";
27     for (int i = 0; i < n; i++){
28         total_wt = total_wt + wt[i];
29         total_tat = total_tat + tat[i];
30         cout << " " << i + 1 << "\t\t" << bt[i] << "\t\t" << wt[i] << "\t\t" << tat[i]
31     }
32     cout << "\nAverage waiting time = " << (float)total_wt / (float)n;
33     cout << "\nAverage turn around time = " << (float)total_tat / (float)n;
34     cout<<endl;
35
36 int main(){
37     int n; cin >> n;
38     int processes[n],burst_time[n];
39     int quantum;
40     cout<<"Enter The Quantum: "<<endl;cin>>quantum;
41     cout << "Enter Burst Times of the Processes: " << endl;
42     for (int i = 0; i < n; i++)
43         processes[i] = i + 1;
44     for (int i = 0; i < n; i++)
45         cin >> burst_time[i];
46     cout << "-----" << endl;
47     cout << " OUTPUT : " << endl;
48     cout << "-----" << endl;
49     findavgTime(processes, n, burst_time, quantum);
50     return 0;}
  
```

```

rajneesh@rajneesh-VirtualBox: ~/Desktop/OS Lab/3$ g++ RoundRobin.cpp -o RR
rajneesh@rajneesh-VirtualBox:~/Desktop/OS Lab/3$ ./RR
3
Enter The Quantum:
2
Enter Burst Times of the Processes:
10 5 8
-----
OUTPUT :
-----
Processes Burst time Waiting time Turn around time
1          10          13          23
2           5          10          15
3           8          13          21
Average waiting time = 12
Average turn around time = 19.6667
rajneesh@rajneesh-VirtualBox:~/Desktop/OS Lab/3$ ./RR
5
Enter The Quantum:
3
Enter Burst Times of the Processes:
10 5 8 4 6
-----
OUTPUT :
-----
Processes Burst time Waiting time Turn around time
1          10          23          33
2           5          15          20
3           8          24          32
4           4          20          24
5           6          21          27
Average waiting time = 20.6
Average turn around time = 27.2
rajneesh@rajneesh-VirtualBox:~/Desktop/OS Lab/3$
  
```