

OS LAB 2

NAME: Aditya Anand

ROLL NO.: 20124009

BRANCH: IT

S NO.	TITLE	DATE OF IMPLEMENTATION	REMARKS
1	Program to implement Round Robin Process of CPU Scheduling	17-01-2022	

ROUND ROBIN CPU SCHEDULING

CRITERIA: Time Quantum

MODE: Pre-emptive

GIVEN: Time Quantum and list of processes with their arrival and burst time.

CODE:

```
#include<bits/stdc++.h>
using namespace std;

class process{
public:
    int id;
    int arrivalTime;
    int burstTime;
    int remainingBurstTime;
    int completionTime;
    int TAT;
    int WT;
    int RT;
    bool isRunning;
};

void round_robin_scheduling(vector<process> &v, int quantum){
    queue<process> q;
    stack<process> s;
    int cur_time = 0;
    int id=0;
    while(true){
        // Add all the processes that have arrived to the ready queue.
        while(id<v.size() && v[id].arrivalTime<=cur_time){
            q.push(v[id]);
            id++;
        }
        // Add the last process from running queue at the end of ready queue if it is not completed.
        if(!s.empty()){
            process p = s.top();
            if(p.remainingBurstTime>0){
                q.push(p);
            }
        }

        // If ready queue is empty => no process to be completed => stop.
        if(q.empty()){
            break;
        }

        // Pick the front process from the ready queue for processing by the CPU.
        process rning_proc = q.front();
        q.pop();
        // Store the Response time for a process the first time it reaches the CPU.
        if(rning_proc.isRunning==false){
```

```

        v[rning_proc.id].RT = cur_time-v[rning_proc.id].arrivalTime;
        rning_proc.isRunning=true;
    }
    // If the remaining burst time > quantum, the process is not complete (CONTEXT SWITCHING)
    if(rning_proc.remainingBurstTime>=quantum){
        rning_proc.remainingBurstTime-=quantum;
        cur_time+=quantum;
    }
    // Process is complete
    else{
        cur_time+=rning_proc.remainingBurstTime;
        rning_proc.remainingBurstTime=0;
    }
    s.push(rning_proc);
    // If process is complete, store the completion time for the process
    if(rning_proc.remainingBurstTime==0){
        v[rning_proc.id].completionTime = cur_time;
    }

}
return;
}

```

```

int main(){

    cout<<"ROUND ROBIN CPU SCHEDULING ALGORITHM C++ IMPLEMENTATION\n";
    cout<<"Name: Aditya Anand\tRoll No.:20124009\t Branch: IT\n\n\n";

    int quantum = 0;
    int n=0;
    cout<<"Enter the number of processes: ";
    cin>>n;
    cout<<"Enter value of time quantum: ";
    cin>>quantum;

    cout<<"Enter the arrival times and burst times of "<<n<<" processes: \n";

    vector<process> v(n);
    for(int i=0; i<n; i++){
        cin>>v[i].arrivalTime>>v[i].burstTime;
        v[i].id = i;
        v[i].isRunning = false;
        v[i].remainingBurstTime = v[i].burstTime;
    }

    round_robin_scheduling(v, quantum);
    int t_TAT=0;
    int t_CT=0;
    for(int i=0; i<n; i++){
        v[i].TAT = v[i].completionTime-v[i].arrivalTime;
        v[i].WT = v[i].TAT-v[i].burstTime;
    }
}

```

```

        t_TAT+=v[i].TAT;
        t_CT+=v[i].completionTime;
    }

    for(auto p:v){
        cout<<"Process: "<<p.id<<"\tArrival Time:"<<p.arrivalTime<<"\tBurst
Time:"<<p.burstTime<<"\tCompletion Time:"<<p.completionTime;
        cout<<"\tTurn Around Time:"<<p.TAT<<"\tWaiting Time:"<<p.WT<<"\tResponse Time:"<<p.RT<<"\n";
    }

    cout<<"\nAverage Turn Around Time: "<<(float)((1.0*t_TAT)/(1.0*n))<<"\n";
    cout<<"\nAverage Completion Time: "<<(float)((1.0*t_CT)/(1.0*n))<<"\n";

    return 0;
}

```

RESULT:

```

PS C:\Users\beadi\Desktop\OS\Assignment 2> cd "c:\Users\beadi\Desktop\OS\Assignment 2\" ; if ($?) { g++ RoundRobin.cpp -o RoundRobin } ; if ($?) { .\RoundRobin }
ROUND ROBIN CPU SCHEDULING ALGORITHM C++ IMPLEMENTATION
Name: Aditya Anand      Roll No.:20124009      Branch: IT

```

```

Enter the number of processes: 4
Enter value of time quantum: 2
Enter the arrival times and burst times of 4 processes:

```

```

0 5
1 4
2 2
4 1

```

Process: 0	Arrival Time:0	Burst Time:5	Completion Time:12	Turn Around Time:12	Waiting Time:7	Response Time:0
Process: 1	Arrival Time:1	Burst Time:4	Completion Time:11	Turn Around Time:10	Waiting Time:6	Response Time:1
Process: 2	Arrival Time:2	Burst Time:2	Completion Time:6	Turn Around Time:4	Waiting Time:2	Response Time:2
Process: 3	Arrival Time:4	Burst Time:1	Completion Time:9	Turn Around Time:5	Waiting Time:4	Response Time:4

Average Turn Around Time: 7.75

Average Completion Time: 9.5