**Number of the Experiment :** 10

**Name of the Experiment :** Implementation of Round Robin Scheduling Algorithm using C program.

**Date of Performance :**

**Date of Submission :**

**Aim and Objectives :** To learn about Round-robin Algorithm and Implement it using a C program.

**Theory :** In Round Robin Scheduling,

* CPU is assigned to the process on the basis of FCFS for a fixed amount of time.
* This fixed amount of time is called as time quantum or time slice.
* After the time quantum expires, the running process is preempted and sent to the ready queue.
* Then, the processor is assigned to the next arrived process.
* It is always preemptive in nature.

**Source Code :**

#include<stdio.h>

int main()

{

int count, j, n, time, remain, flag=0,time\_quantum;

int wait\_time=0, turnaround\_time=0, at[10], bt[10], rt[10];

printf("Enter Total Process:\t ");

scanf("%d",&n);

remain=n;

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

{

printf("Enter Arrival Time and Burst Time for Process Process Number %d :",count+1);

scanf("%d",&at[count]);

scanf("%d",&bt[count]);

rt[count] = bt[count];

}

printf("Enter Time Quantum:\t");

scanf("%d",&time\_quantum);

printf("\n\nProcess\t|Turnaround Time|Waiting Time\n\n");

for( time=0, count=0; remain!=0; )

{

if( rt[count] <= time\_quantum && rt[count]>0)

{

time+ = rt[count];

rt[count] = 0;

flag = 1;

}

else if( rt[count] > 0)

{

rt[count]-= time\_quantum;

time+= time\_quantum;

}

if(rt[count] == 0 && flag == 1)

{

remain--;

printf("P[%d]\t|\t%d\t|\t%d\n",count+1,time-at[count],time-at[count]-bt[count]);

wait\_time+= time-at[count]-bt[count];

turnaround\_time+= time-at[count];

flag = 0;

}

if(count == n-1)

count = 0;

else if(at[count+1] <= time)

count++;

else

count = 0;

}

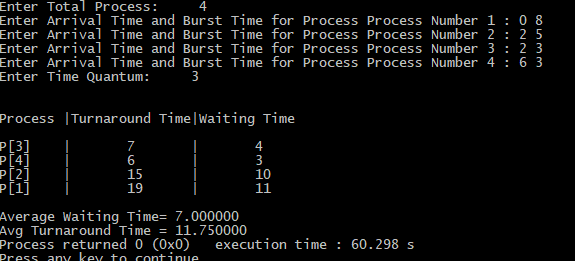
printf("\nAverage Waiting Time= %f\n",wait\_time\*1.0/n);

printf("Avg Turnaround Time = %f",turnaround\_time\*1.0/n);

return 0;

}

**Output :**



**Discussion :** Round-robin is one of the algorithms employed by process and network schedulers in computing. As the term is generally used, time slices (also known as time quanta) are assigned to each process in equal portions and in circular order, handling all processes without priority.