**WEEK -4**

**AIM:-**

To implement best CPU scheduling algorithm (Round Robin Scheduling algorithm)

**Description:-**

 Round robin scheduling is the preemptive scheduling in which every process get executed in a cyclic way, i.e. in this a particular time slice is allotted to each process which is known as time quantum. Every process, which is present in the queue for processing, CPU is assigned to that process for that time quantum. Now, if the execution of the process gets completed in that time quantum, then the process will get terminate otherwise the process will again go to the ready queue, and the previous process will wait for the turn to complete its execution.

**Algorithm:-**

INPUT :- number of processes ,The Execution times(Burst times)and arrival times for the processes

OUTPUT:-turn around time ,waiting time of each process ,average waiting time

**Libraries Used:-**

Header Files:-stdio.h.

**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;

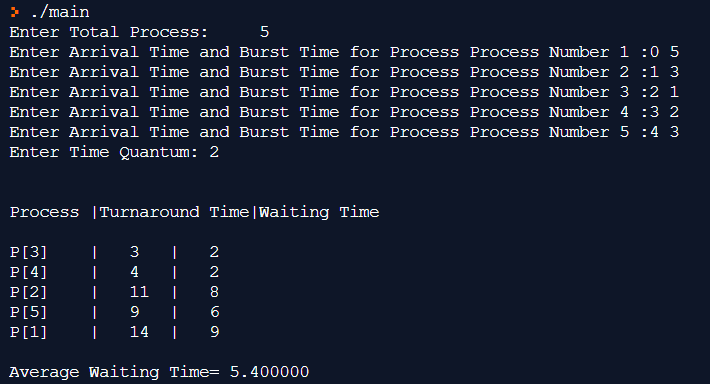
}

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

return 0;

}

**Output:-**



**Observations:-**

Each process is served by the CPU for a fixed time quantum, so all processes are given the same priority.

Starvation doesn't occur because, for each round robin cycle, every process is given a fixed time to execute. No process is left behind.

Also here one cannot assign priority to any process which can be a drawback.