Lab Report NO : 10

Lab report Name: Implementation Roud Robin algorithm.

Theory :

Round robin is a CPU scheduling algorithm that is designed especially for time sharing systems. It is more like a FCFS scheduling algorithm with one change that in Round Robin processes are bounded with a quantum time size. A small unit of time is known as Time Quantum or Time Slice. Time quantum can range from 10 to 100 milliseconds. CPU treat ready queue as a circular queue for executing the processes with given time slice. It follows preemptive approach because fixed time are allocated to processes. The only disadvantage of it is overhead of context switching.

What we need to calculate?

Completion Time is the time required by the process to complete its execution

Turnaround Time is the time interval between the submission of a process and its completion.

Turnaround Time = completion of a process – submission of a process

Waiting Time is the difference between turnaround time and burst time

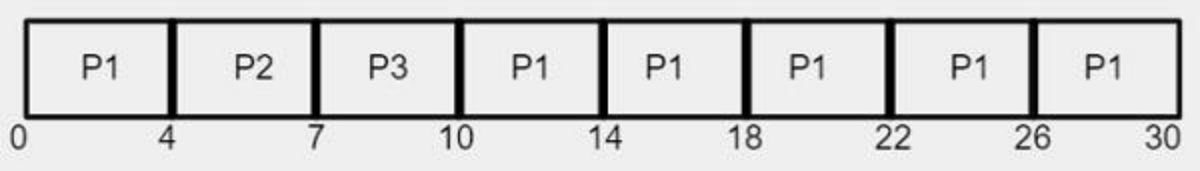
Waiting Time = turnaround time – burst time

Example:

We are given with 3 processes P1, P2 and P3 with their corresponding burst time as 24, 3 and 3

Process	Burst Time
P1	24
P2	3
P3	3

Since the time quantum is of 4 milliseconds, process P1 gets the first 4 milliseconds but it requires another 20 millisecond to complete its execution but CPU will preempt it after the first time quantum and CPU will be allocated to the next process P2. As shown in the table, Process P2 requires only 3 milliseconds to complete its execution so CPU will be allocated for time quantum of 3 milliseconds only instead of 4 milliseconds.



Using the Gantt chart, Average waiting time is calculated as given below −

Average waiting time = 17/3 = 5.66 milliseconds

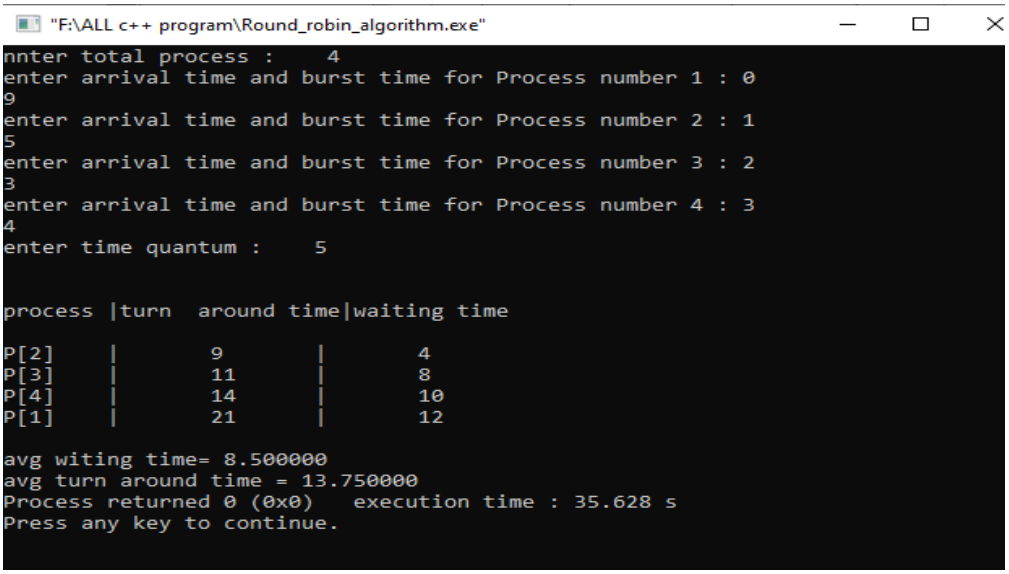
Implementation round robin scheduling algorithm in c

```
#include<stdio.h>

int main()
{
   int at[10],bt[10],rt[10],wt=0,tat=0;
   int cnt,j,x,t,rem,flg=0,time_quant;

   printf("nnter total process :\t ");
   scanf("%d",&x);
   rem=x;
   for(cnt=0; cnt<x; cnt++)
   {
      printf("enter arrival time and burst time for Process number %d : ",cnt+1);
      scanf("%d",&at[cnt]);

      scanf("%d",&bt[cnt]);
      rt[cnt]=bt[cnt];
   }
   printf("enter time quantum :\t");
   scanf("%d",&time_quant);
```

```
printf("\n\nprocess\t|turn  around time|waiting time\n\n");
for(t=0,cnt=0; rem!=0;)
{
   if(rt[cnt]<=time_quant && rt[cnt]>0)
   {
      t+=rt[cnt];
      rt[cnt]=0;
      flg=1;
   }

   else if(rt[cnt]>0)
   {
      rt[cnt]-=time_quant;
      t+=time_quant;
   }
   if(rt[cnt]==0 && flg==1)
   {
      rem--;
      printf("P[%d]\t|\t%d\t|\t%d\n",cnt+1,t-at[cnt],t-at[cnt]-bt[cnt]);
      wt+=t-at[cnt]-bt[cnt];

      tat+=t-at[cnt];
      flg=0;
   }

   if(cnt==x-1)
      cnt=0;
   else if(at[cnt+1]<=t)
      cnt++;
   else
      cnt=0;
```

```
    }

    printf("\navg witing time= %f\n",wt*1.0/x);

    printf("avg turn around time = %f",tat*1.0/x);
return 0;
}
```

Output:



Discussion :

Round robin is a cpu Scheduling Algorith. where each process is assigned a fixed time slot in a cyclic way.

1.It is simple, easy to implement, and starvation-free as all processes get fair share of CPU

2.One of the most commonly used technique in CPU scheduling as a core.

3.It is preemptive as processes are assigned CPU only for a fixed slice of time at most.

4.The disadvantage of it is more overhead of context switching.