
EX NO: 4D	ROUND ROBIN – SCHEDULING ALGORITHM
DATE:	

AIM:

To write a program to implement CPU scheduling for Round Robin Scheduling.

ALGORITHM:

1. Get the number of process and their burst time.
2. Initialize the array for Round Robin circular queue as '0'.
3. The burst time of each process is divided and the quotients are stored on the round Robin array.
4. According to the array value the waiting time for each process and the average time are calculated as line the other scheduling.
5. The waiting time for each process and average times are displayed.
6. According to the array value the waiting time for each process and the average time are calculated as line the other scheduling.

PROGRAM:

```
#include<stdio.h>
void main()
{
int ts,pid[10],need[10],wt[10],tat[10],i,j,n,n1;
int bt[10],flag[10],ttat=0,twt=0;
float awt,atat;
printf("\n ROUND ROBIN SCHEDULING \n");
printf("Enter the number of processors \n");
scanf("%d",&n);
n1=n;
printf("\n Enter the Timeslice \n");
scanf("%d",&ts);
for(i=1;i<=n;i++)
{
printf("\n Enter the process ID %d",i);
scanf("%d",&pid[i]);
printf(" Enter the Burst Time for the process");
scanf("%d",&bt[i]);
need[i]=bt[i];
}
for(i=1;i<=n;i++)
{
flag[i]=1;
wt[i]=0;
}
while(n!=0)
{
for(i=1;i<=n;i++)
{
if(need[i]>=ts)
{
for(j=1;j<=n;j++)
{
if((i!=j)&&(flag[i]==1)&&(need[j]!=0))
wt[j]+=ts;
}
need[i]=0;
n--;
flag[i]=0;
}
}
twt=0;
ttat=0;
for(i=1;i<=n1;i++)
{
tat[i]=wt[i]+bt[i];
twt=twt+wt[i];
ttat=ttat+tat[i];
}
}
```

```
awt=(float)twt/n1;
```

```
atat=(float)ttat/n1;
```

```

printf("\n\n ROUND ROBIN SCHEDULING ALGORITHM \n\n");
printf("\n Process \t Process ID \t BurstTime \t Waiting Time \t Turnaround Time \n");
for(i=1;i<=n1;i++)
{
printf("\n %5d \t %5d \t %5d \t %5d \t %5d \n",i,pid[i],bt[i],wt[i],tat[i]);
}
printf("\n The average Waiting Time=%4.2f",awt);
printf("\n The average Turn around Time=%4.2f",atat);
}

```

OUTPUT:

```

mohamedinam@Mohamed-Inam-PC: ~
mohamedinam@Mohamed-Inam-PC:~$ gcc rr.c -o rr
mohamedinam@Mohamed-Inam-PC:~$ ./rr

ROUND ROBIN SCHEDULING
Enter the number of processors :
4

Enter the Timeslice :
5

Enter the process ID 1 : 5
Enter the Burst Time for the process10

Enter the process ID 2 : 6
Enter the Burst Time for the process15

Enter the process ID 3 : 7
Enter the Burst Time for the process20

Enter the process ID 4 : 8
Enter the Burst Time for the process25

ROUND ROBIN SCHEDULING ALGORITHM

```

Process	Process ID	BurstTime	Waiting Time	Turnaround Time
1	5	10	15	25
2	6	15	25	40
3	7	20	25	45
4	8	25	20	45

```

The average Waiting Time=4.2f
The average Turn around Time=4.2f

```

RESULT:

Thus the Round Robin scheduling ⁵ program was executed and verified successfully.
