Ex. No.: 6d)
Date 06/02/25

ROUND ROBIN SCHEDULING

Aim:

To implement the Round Robin (RR) scheduling technique

Algorithm:

1. Declare the structure and its elements.
2. Get number of processes and Time quantum as input from the user.
3. Read the process name, arrival time and burst time
4. Create an array **rem_bt[]** to keep track of remaining burst time of processes which is initially copy of bt[] (burst times array)
5. Create another array **wt[]** to store waiting times of processes. Initialize this array as 0. 6. Initialize time : t = 0
7. Keep traversing the all processes while all processes are not done. Do following for i'th process if it is not done yet.
   a- If rem_bt[i] > quantum
   (i) t = t + quantum
   (ii) bt_rem[i] -= quantum;
   b- Else // Last cycle for this process
   (i) t = t + bt_rem[i];
   (ii) wt[i] = t - bt[i]
   (iii) bt_rem[i] = 0; // This process is over
8. Calculate the waiting time and turnaround time for each process.
9. Calculate the average waiting time and average turnaround time.
10. Display the results.

Program Code:

```
#include <stdio.h>
int main()
{
    int n;
    Printf("Enter total no. of process: ");
    scanf("%d", &n);
    int wait=0, turaro=0, arr[n], burst[n],
    temp[n];
    int x=n;
for(int i=0; i<n; i++)
{ Printf("enter details %d\n", i+1);
    Printf("Arrival Time: ");
    scanf("%d", &arr[i]);
```

```
        printf("Burst time : ");
        scanf("%d", &Burst[i]);
    }
    int time_Quant;
    Printf("Enter time Quant: ");
    scanf("%d", &time_Quant);
    int total=0, counter=0, i;
    Printf("process ID  Burst Time  turn around time,
            Waiting Time\n");
    for (total=0; i=0, x!=0;)
    {
      if (temp[i] <= time_Quant && temp[i]>0)
        {
          total = total + temp[i];
          temp[i] =0;
          counter=1;
        }
      if (temp[i]==0 && counter==1)
      {
        x--;
        printf("\n process No%d\t\t%d\t\t\t%d\t\t\t%d", i+1,
burst[i], total-arr[i], total-arr[i]-burst[i]);
        wait= wait + total -arr[i] - burst[i];
        turaro = total - arr[i];
        counter=0;
      }
      if (i==n-1)
        i=0;
      else if (arr[i+1] <= total)
        i++;
```

```
        else
            i=0;
    }
    float avg_w = (float) wait/n;
    float avg_t = (float) turnaround/n;
    printf("\n average Waiting time: %.f", avg_w);
    printf("\n average Turnaround Time: %.f", avg_t);
    return 0;
}
```

OUTPUT:

Enter total no. of. process : 3

Enter Details of process : 1

Arrival Time : 0

Burst Time : 4

Enter Details of process 2

Arrival Time : 1

Burst time : 7

Enter Details of process 3

Arrival Time : 2

Burst Time : 5

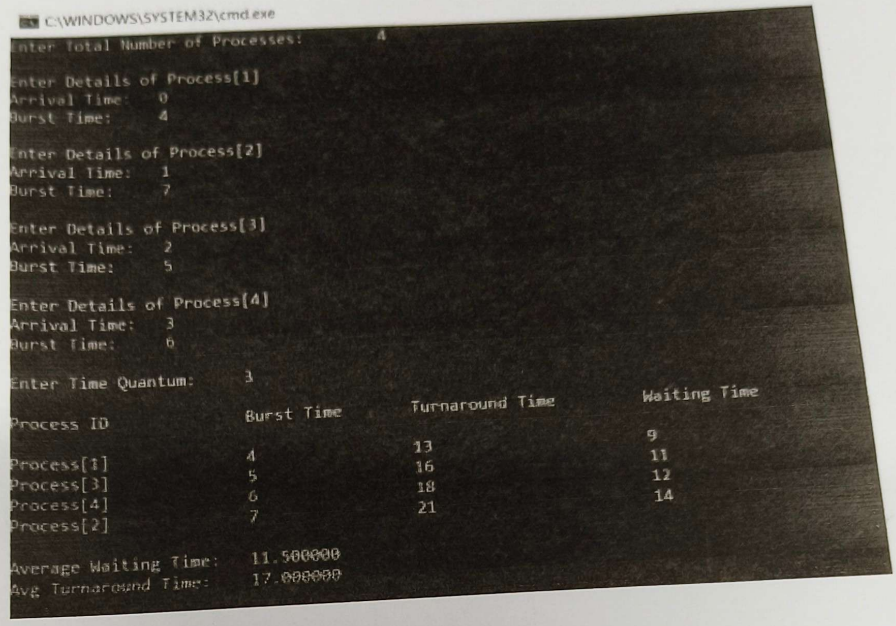
Enter Time Quant : 2

Process ID	Burst Time	TurnAround Time	Waiting time
1	4	8	4
3	5	13	8
2	7	15	8

Avg. Waiting Time : 6.66 ms

Avg. Turn around time : 12.00 ms

Sample Output:



Result:

Hence the Round Robin scheduling has been implemented and Executed Successfully.