What Does Round Robin Scheduling (RRS) Mean?

Round robin scheduling (RRS) is a job-scheduling algorithm that is considered to be very fair, as it uses time slices that are assigned to each process in the queue or line. Each process is then allowed to use the CPU for a given amount of time, and if it does not finish within the allotted time, it is preempted and then moved at the back of the line so that the next process in line is able to use the CPU for the same amount of time.

In computer operation, one method of having different program process take turns using the resources of the computer is to limit each process to a certain short time period, then suspending that process to give another process a turn (or "time-slice"). This is often described as round-robin process scheduling.

ROUND ROBIN SCHEDULING

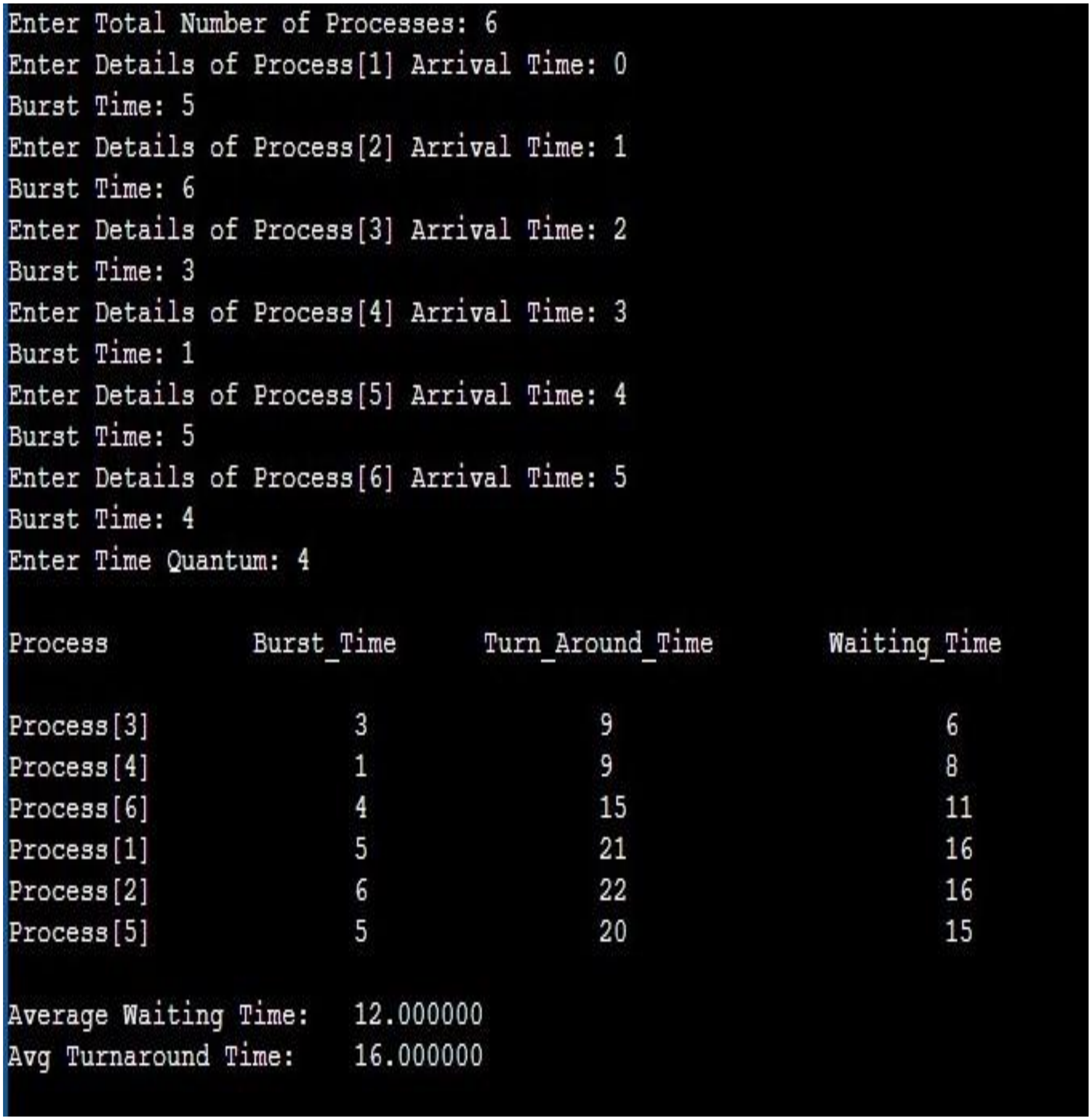
Round Robin is the preemptive process scheduling algorithm. Each process is provided a fix time to execute, it is called a quantum. Once a process is executed for a given time period, it is preempted and other process executes for a given time period. Context switching is used to save states of preempted processes.

CODE

```
#include<stdio.h>


int main()

{

    int i, limit, total = 0, x, counter = 0, time_quantum;

    int wait_time = 0, turnaround_time = 0, arrival_time[10], burst_time[10], temp[10];

    float average_wait_time, average_turnaround_time;

    printf("Enter Total Number of Processes: ");

    scanf("%d", &limit);

    x = limit;

    for(i = 0; i < limit; i++)

    {

        printf("Enter Details of Process[%d] ", i + 1);


        printf("Arrival Time: ");


        scanf("%d", &arrival_time[i]);


        printf("Burst Time: ");


        scanf("%d", &burst_time[i]);
        temp[i] = burst_time[i];
    }

    printf("Enter Time Quantum: ");
    scanf("%d", &time_quantum);
    printf("\nProcess\t\t Burst_Time \t Turn_Around_Time \t Waiting_Time \n");
    for(total = 0, i = 0; x != 0;)
    {
        if(temp[i] <= time_quantum && temp[i] > 0)
        {
            total = total + temp[i];
            temp[i] = 0;
            counter = 1;
        }
        else if(temp[i] > 0)
        {
            temp[i] = temp[i] - time_quantum;
            total = total + time_quantum;
        }
        if(temp[i] == 0 && counter == 1)
        {
            x--;
            printf("\nProcess[%d]\t\t%d\t\t   %d\t\t\t   %d", i + 1, burst_time[i], total -
arrival_time[i], total - arrival_time[i] - burst_time[i]);
```

```
                wait_time = wait_time + total - arrival_time[i] - burst_time[i];
                turnaround_time = turnaround_time + total - arrival_time[i];
                counter = 0;
            }
            if(i == limit - 1)
            {
                i = 0;
            }
            else if(arrival_time[i + 1] <= total)
            {
                i++;
            }
            else
            {
                i = 0;
            }
        }

        average_wait_time = wait_time * 1.0 / limit;
        average_turnaround_time = turnaround_time * 1.0 / limit;
        printf("\n\nAverage Waiting Time:\t%f", average_wait_time);
        printf("\nAvg Turnaround Time:\t%f\n", average_turnaround_time);
        return 0;
    }
```

OUTPUT



```
Enter Total Number of Processes: 6
Enter Details of Process[1] Arrival Time: 0
Burst Time: 5
Enter Details of Process[2] Arrival Time: 1
Burst Time: 6
Enter Details of Process[3] Arrival Time: 2
Burst Time: 3
Enter Details of Process[4] Arrival Time: 3
Burst Time: 1
Enter Details of Process[5] Arrival Time: 4
Burst Time: 5
Enter Details of Process[6] Arrival Time: 5
Burst Time: 4
Enter Time Quantum: 4

Process          Burst_Time      Turn_Around_Time        Waiting_Time

Process[3]              3                9                       6
Process[4]              1                9                       8
Process[6]              4                15                      11
Process[1]              5                21                      16
Process[2]              6                22                      16
Process[5]              5                20                      15

Average Waiting Time:   12.000000
Avg Turnaround Time:    16.000000
```