

Round-Robin.cpp

```
// C++ program for implementation of RR scheduling
#include <iostream>
#include <stdio>
#include <stdlib>
#define NUM_PROC      1000
#define SWITCH_OVERHEAD 1
#define TIME_QUANTUM  2

using namespace std;

// Function to find the waiting time for all
// processes
int findWaitingTime(int n, int bt[], int wt[], int rt[], int quantum)
{
    // Make a copy of burst times bt[] to store remaining
    // burst times.
    int rem_bt[n];
    for (int i = 0 ; i < n ; i++)
        rem_bt[i] = bt[i];

    int t = 0; // Current time

    // Keep traversing processes in round robin manner
    // until all of them are not done.
    while (1) {
        bool done = true;

        // Traverse all processes one by one repeatedly
        for (int i = 0 ; i < n; i++) {
            if (rem_bt[i] == bt[i]) // first running quantum
                rt[i] = t + SWITCH_OVERHEAD; // response time
            // If burst time of a process is greater than 0
            // then only need to process further
            if (rem_bt[i] > 0) {
                done = false; // There is a pending process
                printf("%5d: SWITCHING  %5d\n", t, SWITCH_OVERHEAD);
                t = t + SWITCH_OVERHEAD;

                if (rem_bt[i] > quantum) {
                    printf("%5d: proc(%3d)  %5d\n", t, i, quantum);
                    // Increase the value of t i.e. shows
                    // how much time a process has been processed
                    t += quantum;

                    // Decrease the burst_time of current process
                    // by quantum
                    rem_bt[i] -= quantum;
                }
            }
        }
    }
}
```

```

        // If burst time is smaller than or equal to
        // quantum. Last cycle for this process
        else {
            printf("%5d: proc(%3d)  %5d\n", t, i, rem_bt[i]);
            // Increase the value of t i.e. shows
            // how much time a process has been processed
            t = t + rem_bt[i];

            // Waiting time is current time minus time
            // used by this process
            wt[i] = t - bt[i];

            // As the process gets fully executed
            // make its remaining burst time = 0
            rem_bt[i] = 0;
        }
    } // if
} // for

// If all processes are done
if (done == true)
    break;
}

return t; // Completion time
}

// Function to calculate turn around time
void findTurnAroundTime(int n, int bt[], int wt[], int tat[])
{
    // calculating turnaround time by adding
    // bt[i] + wt[i]
    for (int i = 0; i < n ; i++)
        tat[i] = bt[i] + wt[i];
}

// Function to calculate average time
void findavgTime(int n, int bt[], int quantum)
{
    int wt[n], tat[n], total_wt = 0, total_tat = 0;
    int rt[n], total_rt = 0; // response time
    int ct; // completion time

    // Function to find waiting time of all processes
    ct = findWaitingTime(n, bt, wt, rt, quantum);

    // Function to find turn around time for all processes
    findTurnAroundTime(n, bt, wt, tat);

    printf("Switching Time = %d\n", SWITCH_OVERHEAD);

```

```

        printf("Time Quantum   = %d\n", TIME_QUANTUM);

    printf("ProcessID  Arrival Time  Burst Time  Waiting Time  Turnaround Time\n");
    printf("Response Time\n");
    for (int i = 0 ; i < n ; i++) {
        total_wt = total_wt + wt[i];
        total_tat = total_tat + tat[i];
        total_rt = total_rt + rt[i];
        int compl_time = tat[i] + 0;
        printf("    %5d    %7d    %7d    %7d    %7d    %7d\n",
            i, 0, bt[i], wt[i], tat[i], rt[i]);
    }
    printf("Avg. Waiting Time    = %9.3f\n", (float)total_wt / (float)n);
    printf("Avg. Turnaround Time = %9.3f\n", (float)total_tat / (float)n);
    printf("Avg. Response Time   = %9.3f\n", (float)total_rt / (float)n);
    printf("Completion Time      = %6d\n", ct);
    printf("Throughput(#Jobs/Time)= %9.3f\n", (float)n / (float)ct);
}

int inputData(int burst_time[])
{
    int    i = 0;
    int    num;
    do {
        num = scanf("%d", &burst_time[i]);
        if (num <= 0)    // End-of-file or zero data
            break;
        i++;
    } while (1);
    return i;
}

// Driver code
int main(int argc, char *argv[])
{
    // Burst time of all processes
    int burst_time[NUM_PROC];
    int n;

    n = inputData(burst_time);

    // Time quantum
    int quantum;
    if (argc == 1)
        quantum = TIME_QUANTUM; // default
    else
        quantum = atoi(argv[1]);
    findavgTime(n, burst_time, quantum);
    return 0;
}

```

Sample-rr.dat

```
8
3
9
7
```

time quantum = 2

```
(base) 202020827@cslinux:~/os/round-robin$ ./Round-Robin 2 < sample-rr.dat
0: SWITCHING 1
1: proc( 0) 2
3: SWITCHING 1
4: proc( 1) 2
6: SWITCHING 1
7: proc( 2) 2
9: SWITCHING 1
10: proc( 3) 2
12: SWITCHING 1
13: proc( 0) 2
15: SWITCHING 1
16: proc( 1) 1
17: SWITCHING 1
18: proc( 2) 2
20: SWITCHING 1
21: proc( 3) 2
23: SWITCHING 1
24: proc( 0) 2
26: SWITCHING 1
27: proc( 2) 2
29: SWITCHING 1
30: proc( 3) 2
32: SWITCHING 1
33: proc( 0) 2
35: SWITCHING 1
36: proc( 2) 2
38: SWITCHING 1
39: proc( 3) 1
40: SWITCHING 1
41: proc( 2) 1
Switching Time = 1
Time Quantum = 2
ProcessID  Arrival Time  Burst Time  Waiting Time  Turnaround Time  Response Time
0          0           8           27           35             1
1          0           3           14           17             4
2          0           9           33           42             7
3          0           7           33           40            10
Avg. Waiting Time = 26.750
Avg. Turnaround Time = 33.500
Avg. Response Time = 5.500
Completion Time = 42
Throughput(#Jobs/Time)= 0.095
```

time quantum = 4

```

● (base) 202020827@cslinux:~/os/round-robin$ ./Round-Robin 4 < sample-rr.dat
  0: SWITCHING      1
  1: proc( 0)       4
  5: SWITCHING      1
  6: proc( 1)       3
  9: SWITCHING      1
 10: proc( 2)       4
 14: SWITCHING      1
 15: proc( 3)       4
 19: SWITCHING      1
 20: proc( 0)       4
 24: SWITCHING      1
 25: proc( 2)       4
 29: SWITCHING      1
 30: proc( 3)       3
 33: SWITCHING      1
 34: proc( 2)       1
Switching Time = 1
Time Quantum   = 2
ProcessID  Arrival Time  Burst Time  Waiting Time  Turnaround Time  Response Time
      0           0         8          16           24           1
      1           0         3           6           9           6
      2           0         9          26          35          10
      3           0         7          26          33          15
Avg. Waiting Time = 18.500
Avg. Turnaround Time = 25.250
Avg. Response Time = 8.000
Completion Time = 35
Throughput(#jobs/Time)= 0.114

```

time quantum = 6

```

● (base) 202020827@cslinux:~/os/round-robin$ ./Round-Robin 6 < sample-rr.dat
  0: SWITCHING      1
  1: proc( 0)       6
  7: SWITCHING      1
  8: proc( 1)       3
 11: SWITCHING      1
 12: proc( 2)       6
 18: SWITCHING      1
 19: proc( 3)       6
 25: SWITCHING      1
 26: proc( 0)       2
 28: SWITCHING      1
 29: proc( 2)       3
 32: SWITCHING      1
 33: proc( 3)       1
Switching Time = 1
Time Quantum   = 2
ProcessID  Arrival Time  Burst Time  Waiting Time  Turnaround Time  Response Time
      0           0         8          20           28           1
      1           0         3           8           11           8
      2           0         9          23           32          12
      3           0         7          27           34          19
Avg. Waiting Time = 19.500
Avg. Turnaround Time = 26.250
Avg. Response Time = 10.000
Completion Time = 34
Throughput(#jobs/Time)= 0.118

```

time quantum = 8

```

● (base) 202020827@cslinux:~/os/round-robin$ ./Round-Robin 8 < sample-rr.dat
 0: SWITCHING      1
 1: proc( 0)       8
 9: SWITCHING      1
10: proc( 1)       3
13: SWITCHING      1
14: proc( 2)       8
22: SWITCHING      1
23: proc( 3)       7
30: SWITCHING      1
31: proc( 2)       1
Switching Time = 1
Time Quantum   = 2
ProcessID  Arrival Time  Burst Time  Waiting Time  Turnaround Time  Response Time
      0           0         8           1           9           1
      1           0         3          10          13          10
      2           0         9          23          32          14
      3           0         7          23          30          23
Avg. Waiting Time = 14.250
Avg. Turnaround Time = 21.000
Avg. Response Time = 12.000
Completion Time   = 32
Throughput(#Jobs/Time)= 0.125

```

1. 문맥교환 시간 : SWITCH_OVERHEAD 매크로 값 = 1 단위 시간

2. Time Quantum 변화에 따른 비교

Time Quantum	Avg. Response Time	Avg. Waiting Time	Throughput (#Jobs/Time)	문맥교환 회수
2	5.50	26.75	4 / 42 \approx 0.095	15
4	8.00	18.50	4 / 35 \approx 0.114	8
6	10.00	19.50	4 / 34 \approx 0.118	7
8	12.00	14.25	4 / 32 = 0.125	5

3. 증가·감소 경향

증가하는 척도

평균 Response Time

Throughput

감소하는 척도

평균 Waiting Time (전체적으로 ↓)

문맥교환 회수

4. CPU 효율 : 총 실행 시간 합 = 8 + 3 + 9 + 7 = 27

CPU 효율 = (유효 작업 시간) / (총 경과 시간) · 100%

Time Quantum	Completion Time	CPU 효율
2	42	27 / 42 \approx 0.643 → 64.3 %
4	35	27 / 35 \approx 0.771 → 77.1 %
6	34	27 / 34 \approx 0.794 → 79.4 %
8	32	27 / 32 \approx 0.844 → 84.4 %