



Report On

**SIMULATION BASED PROCESS
SCHEDULER**

For Course
Operating Systems
(CSE 316)

BY:	
Registration No	11812851
Name of Student	AVINASH KUMAR
Roll No	51
Section	K18FG
Email ID:	Avirms07@gmail.com
GitHub Link:	https://github.com/thaakur/OS-CA-2-Round-Robin-Process-Scheduler

Submitted To: **NAVNEET KAUR**

Faculty of: **Lovely Professional University**
Jalandhar, Punjab, India

Contents

Serial No	Name	Page No
1.	Problem Description	3
2.	Test Cases	4
3.	Algorithm	7
4.	Time Complexity	9
5.	Boundary Condition	9
6.	Problem Solution Code	10
7.	Revision of Test Cases	13

Problem Description

Problem states that consider n number of processes and design a process management scheduler, which assign a processor to each process on the basis given conditions. Problem says take arrival time of the process and the required burst time of that process as input from the user at run time and calculate the completion time, turnaround time and waiting time of all the process and also calculates the average turnaround time and waiting time of the processes. While calculating the completion time of the process, we have to follow some set of rules.

- The scheduler schedules the processes by interrupting the processor after every 3 units of time and does consider the completion of the process in this iteration. (First iteration)
- The scheduler then checks for the number of processes waiting for the processor and allots the processor to the process but interrupting the processor after every 6 units of time and considers the completion of the process in this iteration. (Second iteration)
- The scheduler after the second iteration checks for the number of processes waiting for the processor and now provides the processor to the process with the least time requirement to go in the terminated state.

Basically we have to implement 2 algorithms for design such type scheduler,

1. Round-Robin
2. Shortest job first

Round-Robin: Round-Robin algorithm having a fixed time quantum to execute each process in this type of scheduling approach every process will execute at a certain time and no process will go to starvation.

Shortest job First: It is a non-primitive scheduling algorithm where it works on the basis of arrival time and burst time it assigned CPU to that very process which required less time to fully execute and then move to next process in this process large process may be go at starvation.

Test Cases

Test Case: 1

Sample Input:

Process	Arrival time	Burst time
P ₁	0	18
P ₂	2	23
P ₃	4	13
P ₄	13	10

Expected Output:

Process	Arrival time	Burst time	Completion time	Turnaround time	Waiting time
P ₁	0	18	47	47	29
P ₂	2	23	64	62	39
P ₃	4	13	22	18	5
P ₄	13	10	32	19	9

Average Turnaround time: 36.5

Average Waiting time: 20.5

Explanation: The time quantum of 1 iteration is 3 unit of time, the time quantum of 2 iteration is 6 unit of time and after 2 iteration provide CPU of that process which required less time to go terminated state.

Gantt chart of Process Completion:

P ₁	P ₂	P ₃	P ₄	P ₁	P ₂	
0	3	9	22	32	47	64

Test Case: 2

Sample Input:

Process	Arrival time	Burst time
P ₁	0	20
P ₂	3	36
P ₃	13	19

P ₄	20	42
P ₅	36	69
P ₆	41	35

Expected Output:

Process	Arrival time	Burst time	Completion time	Turnaround time	Waiting time
P ₁	0	20	26	26	6
P ₂	3	36	75	72	36
P ₃	13	19	45	32	13
P ₄	20	42	152	132	90
P ₅	36	69	221	185	116
P ₆	41	35	110	69	34

Average Turnaround time: 86

Average Waiting time: 49.1667

Explanation: The time quantum of 1 iteration is 3 unit of time, the time quantum of 2 iteration is 6 unit of time and after 2 iteration provide CPU of that process which required less time to go terminated state.

Gantt chart of Process Completion:

P ₁	P ₂	P ₁	P ₃	P ₂	P ₆	P ₄	P ₅	
0	3	9	26	45	75	110	152	221

Test Case: 3

Sample Input:

Process	Arrival time	Burst time
P ₁	0	6
P ₂	3	16
P ₃	5	2
P ₄	9	8
P ₅	10	7

P ₆	22	5
P ₇	14	13
P ₈	26	4
P ₉	17	19
P ₁₀	28	3

Expected Output:

Process	Arrival time	Burst time	Completion time	Turnaround time	Waiting time
P ₁	0	6	14	14	8
P ₂	3	16	51	48	32
P ₃	5	2	11	6	4
P ₄	9	8	29	20	12
P ₅	10	7	21	11	4
P ₆	22	5	41	19	14
P ₇	14	13	64	50	37
P ₈	26	4	36	10	6
P ₉	17	19	83	66	47
P ₁₀	28	3	32	4	1

Average Turnaround time: 24.8

Average Waiting time: 16.5

Explanation: The time quantum of 1 iteration is 3 unit of time, the time quantum of 2 iteration is 6 unit of time and after 2 iteration provide CPU of that process which required less time to go terminated state.

Gantt chart of Process Completion:

P ₁	P ₂	P ₃	P ₁	P ₅	P ₄	P ₁₀	P ₈	P ₆	P ₂	P ₇	P ₉	
0	3	9	11	14	21	29	32	36	41	51	62	83

Algorithm

Step -1: Declare array arival_time[], busrt_time[], completion_time[], waiting_time[], trunaround_time[] and **int** variable size.

Step -2: Take input number of process in size.

Step -3: Repeat for int I = 0,1,2.... Size-1

. Take arrival time and burst time of the process as input.

Step -4: Remaining process = size , time_quantam_i1=3, time_quantam_i2=6;

. current_time=0 and Repeat step -5 to step - 9 while remaining process != 0

Step -5: if(remanning_bt[Process_no]<time_quantam_i1 && remanning_bt
. [Process_no]>=0&¤t_time<9)

current_time+=remanning_bt[Process_no];

remanning_bt[Process_no]=0

indicator = 1

time_quantam_itrations++

remmaning_process--

Step -6: else if(remanning_bt[Process_no]>0&¤t_time<9)

if(time_quantam_itrations==1)

remanning_bt[Process_no]-=time_quantam_i1

time_quantam_itrations++

current_time+=time_quantam_i1

else if(time_quantam_itrations==2)

remanning_bt[Process_no]-=time_quantam_i2

current_time+=time_quantam_i2

Step -7: else if(remanning_bt[Process_no]<9 && remanning_bt

. [Process_no]>=3&¤t_time<9)

current_time+=remanning_bt[Process_no];

remanning_bt[Process_no]=0;

remmaning_process-- , indicator = 1

Step -8: else if(remanning_bt[Process_no]>3&¤t_time<9)

remanning_bt[Process_no]-=time_quantam_i2

current_time+=time_quantam_i2

Step -9: if(remanning_bt[Process_no]==0 && indicator==1)

Remening_proc--

completion_time[Process_no]=current_time

trunaround_time[Process_no]=completion_time[Process_no]-
arival_time[Process_no]

waiting_time[Process_no]=trunaround_time[Process_no]-
bust_time[Process_no]

indicator = 0

Step -10: End of step 4 loop, set:- Process_no=0

Repeat Step -11 to Step - 12 while remaining process != 0

Step -11: Pick a job whose burst time is lesser among all of them.

Step -12: Execute that job and set

completion_time[Process_no]=current_time

trunaround_time[Process_no]=completion_time[Process_no]-
arival_time[Process_no]

waiting_time[Process_no]=trunaround_time[Process_no]-
bust_time[Process_no]

Step -13: End of step 10 loop.

Step -14: Repeat for int I = 0,1,2.... Size-1

. **A)** Calculate average waiting time and average burst time.

. **B)** Print all the information.

Time Complexity

Step -1 and Step -2 having time complexity constant time complexity : $O(1)$.

Step -3 having time complexity no of process : $O(n)$.

Step -4 to Step-9 having time complexity two iteration : $O(m)$.

Step -10 to Step 13 having time complexity : $O(n^2)$.

Step -11 having time complexity no of process : $O(n)$.

Overall Time Complexity is : $O(n^2)$.

Boundary Conditions

1. The first iteration should be follow Round Robin scheduling algorithm with time quantum of 3.
 - Arrival time must be greater and equal to 0
 - If any process having burst time less than 3, then set its completion time is burst time and remove from ready queue.
2. Second iteration also should be follow the Round Robin scheduling algorithm but having time quantum of 6.
 - Arrival time must be greater and equal to 0
 - And also check the ready queue if any process waiting already then execute first that very process.
 - If any process having burst time less than 9, then set its completion time is burst time and remove from ready queue.
3. After the second iteration follow then Shortest Job First scheduling algorithm.
 - Current time quantum must be greater or equals to 9
 - And also check the ready queue if any process waiting already then execute first that very process according to which have less time required to fully execute.
 - While picking up any shortest job to execute must be check that its arrival time equals of greater than current execution time.

Revision of Test Cases

The screenshot shows a Windows desktop with a Python script execution window, a calendar, and a system clock. The Python window displays the following output:

```
Enter the arrival time and burst time of the processes

Process P1
Arrival time = 0
Burst time = 18

Process P2
Arrival time = 2
Burst time = 23

Process P3
Arrival time = 4
Burst time = 13

Process P4
Arrival time = 13
Burst time = 10

Process
Arrival time    Burst time    Completion time    Turnaround Time    waiting time
0               18            18                 0                  0
2               23            27                 0                  0
4               13            0                  0                  0
13              10            0                  0                  0

Average Trunaround time : 0
Average Waiting time : 0

Process exited after 10.66 seconds with return value 0
Press any key to continue . . .
```

The calendar shows March 2020, with the 8th highlighted. The system clock displays 6:10:34 PM on Sunday, March 8, 2020.

Second Revision

The screenshot shows a Windows desktop with a Python script execution window, a calendar, and a system clock. The Python window displays the following output:

```
Enter the arrival time and burst time of the processes

Process P1
Arrival time = 0
Burst time = 18

Process P2
Arrival time = 2
Burst time = 23

Process P3
Arrival time = 4
Burst time = 13

Process P4
Arrival time = 13
Burst time = 10

Process
Arrival time    Burst time    Completion time    Turnaround Time    waiting time
0               18            0                  0                  0
2               23            0                  0                  0
4               13            13                 0                  0
13              10            23                 0                  0

Average Trunaround time : 0
Average Waiting time : 0

Process exited after 8.902 seconds with return value 0
Press any key to continue . . .
```

The calendar shows March 2020, with the 13th highlighted. The system clock displays 10:36:56 PM on Friday, March 13, 2020.

Third Revision :

Enter the arrival time and burst time of the processes

Process P1
Arrival time = 0
Burst time = 18

Process P2
Arrival time = 2
Burst time = 23

Process P3
Arrival time = 4
Burst time = 13

Process P4
Arrival time = 13
Burst time = 10

Process	Arival time	Burst time	Completion time	Turnaround Time	waiting time
P1	0	18	0	0	0
P2	2	23	0	0	0
P3	4	13	22	0	0
P4	13	10	32	0	0

Average Trunaround time : 0
Average Waiting time : 0

Process exited after 65.6 seconds with return value 0
Press any key to continue . . .

Compilation Log:
- Errors: 0
- Warnings: 0
- Output Filename: E:\Python\try.exe
- Output Size: 1.83413887023926 MiB
- Compilation Time: 3.84s

System Clock: 11:57:42 PM, Sunday, March 15, 2020

Fourth Revision :

Enter the arrival time and burst time of the processes

Process P1
Arrival time = 0
Burst time = 18

Process P2
Arrival time = 2
Burst time = 23

Process P3
Arrival time = 4
Burst time = 13

Process P4
Arrival time = 13
Burst time = 10

Process	Arival time	Burst time	Completion time	Turnaround Time	waiting time
P1	0	18	47	0	0
P2	2	23	64	0	0
P3	4	13	22	0	0
P4	13	10	32	0	0

Average Trunaround time : 0
Average Waiting time : 0

Process exited after 11.62 seconds with return value 0
Press any key to continue . . .

Compilation Log:
- Output Size: 1.83413887023926 MiB
- Compilation Time: 3.67s

System Clock: 6:27:09 AM, Thursday, March 19, 2020

Fifth Revision : Successful

(Operating-System-CSE316)\os_code.cpp - [Executing] - Dev-C++ 5.11

Execute Tools AStyle Window Help

TDM-GCC 4.9.2 64-bit Release

```
ce std;
```

C:\Users\Vaibhav\Documents\GitHub\Operating-System-CSE316\os_code.exe

```
Process P2
Arrival time = 2
Burst time = 23

Process P3
Arrival time = 4
Burst time = 13

Process P4
Arrival time = 13
Burst time = 10

The details of time quantum are as follows:

The time quantum for first Iteration is 3.
The time quantum for first Iteration is 6.
After second iteration the Shortest job will assign CPU.

Process      Arrival time    Burst time    Completion time    Turnaround Time    waiting time
P1           0                18            47                 47                 29
P2           2                23            64                 62                 39
P3           4                13            22                 18                 5
P4          13                10            32                 19                 9

Average Trunaround time : 36.5
Average Waiting time : 20.5

-----
Process exited after 11.28 seconds with return value 0
Press any key to continue . . .
```

```
rs: 0
ngs: 0
t Filename: C:\Users\Vaibhav\Documents\GitHub\Operating-System-CSE316\os_code.exe
t Size: 1.83560371398926 MiB
lation Time: 1.81s
```

8:35:33 AM

Saturday, March 21, 2020

March 2020

Su	Mo	Tu	We	Th	Fr	Sa
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31	1	2	3	4
5	6	7	8	9	10	11

Show agenda ^

8:35 AM