

---

---

# Operating System 111 Fall

---

## Homework 3

---

Prof. 蔡文錦

TA. 黃逸弘, 余孟倫, 王麗婷, 王菱君

---

---

# Process Scheduling

1. Shortest Remaining Time First Scheduling (SRTF, Preemptive SJF)
2. Round-Robin(RR)
3. Multilevel Feedback Queue
  - 1<sup>st</sup> Level : Round-Robin
  - 2<sup>nd</sup> Level : First Come First Serve(FCFS)

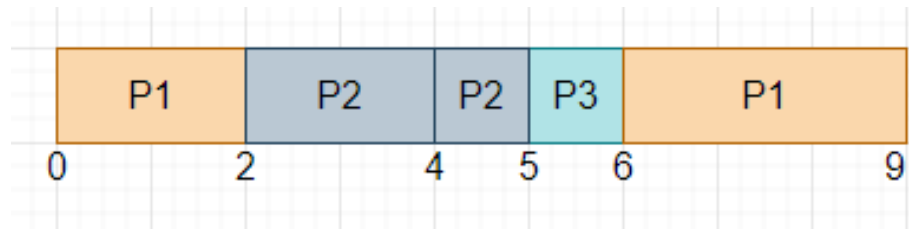
# Scheduling Criteria

Throughput	Number of completed processes per time unit.
Turnaround Time	The amount of time taken to complete a process.
Waiting Time	Total time spent by the process in the ready queue.
Response Time	The time spent when the process is being submission and gets the CPU for the first time.

# Shortest Remaining Time First Scheduling (SRTF)

## Example

Process	Arrival	CPU burst
1	0	5
2	2	3
3	4	1



**Waiting Time**                      **P1 = 4, P2 = 0, P3 = 1**

**Total Waiting Time**            **[ 4 + 0 + 1 ] = 5**

**Turnaround Time**                **P1 = 9, P2 = 3, P3 = 2**

**Total Turnaround time**        **[ 9 + 3 + 2 ] = 14**

# The Format of Input & Output

## Input

3	# Total Number of Processes
0 2 4	# Arrival Time of Each Process
5 3 1	# Burst Time of Each Process

## Output

You should output **four things** :

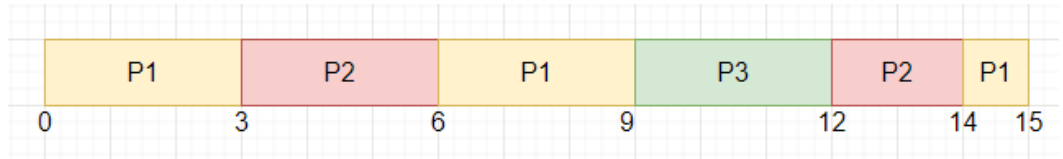
1. Waiting time of each process
2. Turnaround time of each process
3. Total waiting time
4. Total turnaround time

[Click here to see the output format](#)

# Round-Robin

Example with Time Quantum = 3

Process	Arrival	CPU burst
1	0	7
2	2	5
3	4	3



**Waiting Time**                      **P1 = 8, P2 = 7, P3 = 5**

**Total Waiting Time**            **[ 8 + 7 + 5 ] = 20**

**Turnaround Time**                **P1 = 15, P2 = 12, P3 = 8**

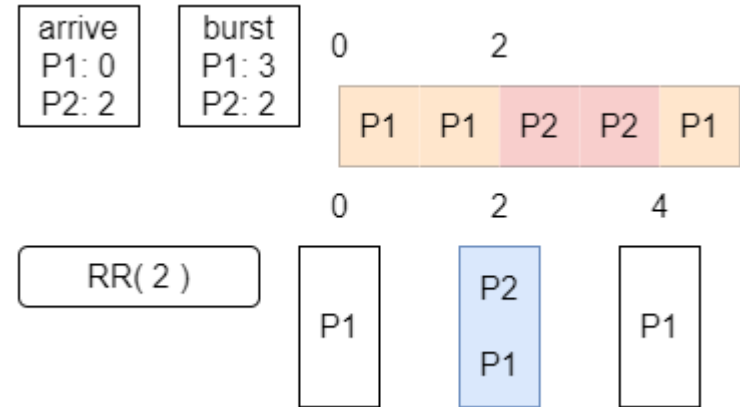
**Total Turnaround time**        **[ 15 + 12 + 8 ] = 35**

# Round-Robin (RR)

Pay attention to this case:

A new process comes while old process just leave CPU because of time quantum.

In this case, CPU will select new process.



# The Format of Input & Output

## Input

3	# Total Number of Processes
0 2 4	# Arrival Time of Each Process
7 5 3	# Burst Time of Each Process
3	# Time Quantum

## Output

You should output **four things** :

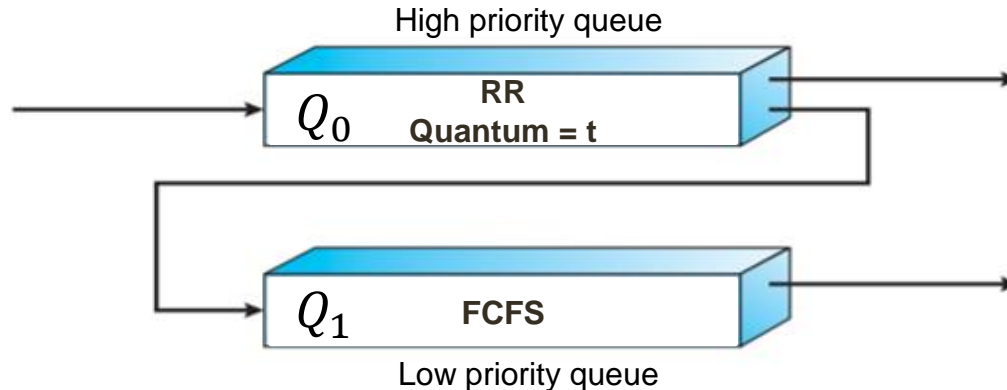
1. Waiting time for each process
2. Turnaround time for each process
3. Total waiting time
4. Total turnaround time

[Click here to see the output format](#)



# Multilevel Feedback Queue (RR+FCFS)

1. Processes in lower priority queue is selected if the higher queues are empty.
2. If a new process comes to higher priority queue, currently executing process in lower priority queue will be preempted by it.
3. When a new job comes, it first enters queue  $Q_0$  which is served as RR. As it gains CPU, job receives "t" time unit.
4. If it doesn't finish in "t" time unit, job is moved to the next queue  $Q_1$  which is served as FCFS.
5. The preempted process will be re-inserted to the end of  $Q_1$ .



# Multilevel Feedback Queue

Example with Time Quantum = 3

Process	Arrival	CPU burst
1	0	7
2	2	4
3	8	1

**Waiting Time**

**P1 = 5, P2 = 4, P3 = 0**

**Total Waiting Time**

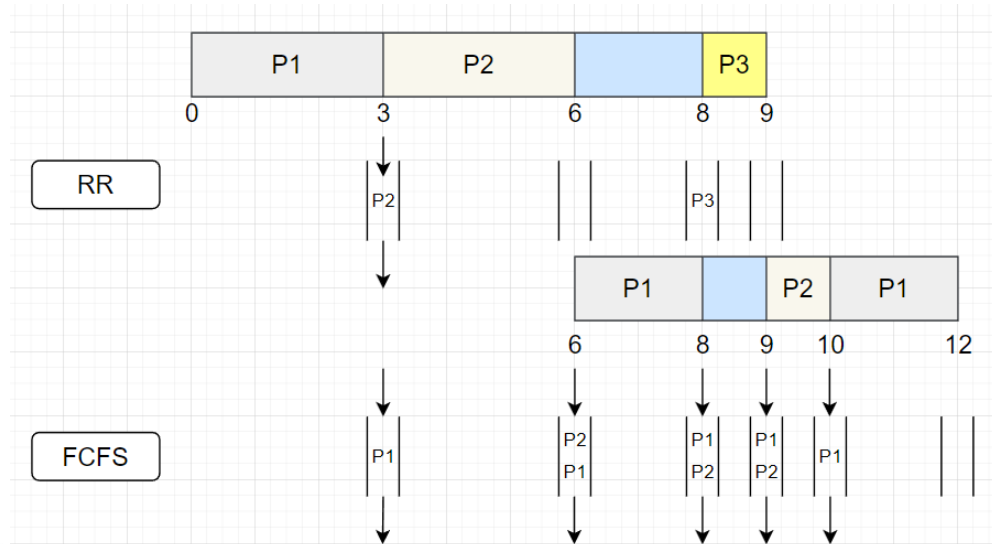
**[ 5 + 4 + 0 ] = 9**

**Turnaround Time**

**P1 = 12, P2 = 8, P3 = 1**

**Total Turnaround time**

**[ 11 + 8 + 1 ] = 20**



# The format of input file & output

## Input

3	# Total Number of Processes
0 2 8	# Arrival Time of Each Process
7 4 1	# Burst Time of Each Process
3	# Time Quantum for RR

## Output

You should output **four things** :

1. Waiting time for each process
2. Turnaround time for each process
3. Total waiting time
4. Total turnaround time

[Click here to see the output format](#)

# The Format of Output (Take SRTF for Example)

```
4 9
0 3
1 2
5
14
```

time.txt

```
Process    Waiting Time    Turnaround Time
P[1]       4                9
P[2]       0                3
P[3]       1                2
Total Waiting Time: 5
Total Turnaround Time: 14
```

Detail Explain

**You just need to print the format as time.txt**

number space number\n

number space number\n

...

number\n number

# Notes

TAs will use “`g++ hw3_1.cpp -o hw3_1`” to compile.

In Linux, after saving the test case in `testcase.txt`, we can use “`./hw3_1 < testcase.txt > student_ans.txt`” to test, and compare it to `answer.txt`.

So, please use **`std::cin`** to get user input in your homework!

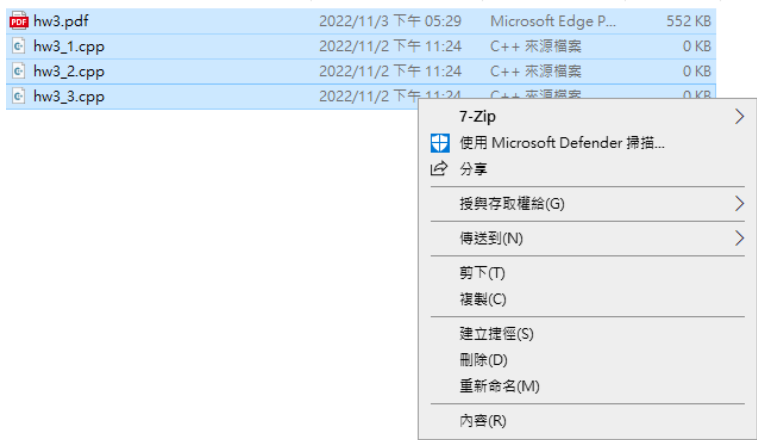
# Grade

Total score : 100pts. Copy will get 0 point!

1. hw3-1, hw3-2, hw3-3 : 30 pts for each question.
2. There will be three test cases for each scheduling algorithm, one is visible, and the others are hidden. You will get 10 pts, if you pass one case.
3. Report : 10 pts
4. The format of Report is in **hw3\_report.docx**. Please **export it to PDF** ( hw3\_report.pdf ) before submitting.
5. If you just use { printf ( "0 7\n5 9\n7 8\n12\n24" ) ; }, you will get 0 pts.
6. Deadline : 2022/11/30 (Wes) 23:59. Late submission will get a **-20% point per day**.

# Requirements

1. You should write codes in **c/c++**.
2. The name of .cpp file must in the form of "**hw3\_1.cpp**" & "**hw3\_2.cpp**" & "**hw3\_3.cpp**".
3. Put all \*.cpp source files and report(\*.pdf) into same compressed file. The type of compressed file must be "**zip**".



# Requirements

The name of your compressed file must have the form of “studentID\_hw3.zip” and **without any folder**.

Incorrect file form : -20 pts

( Including the names of ***compressed file, .cpp files, report file*** )

---

```
studentID_hw3.zip
├── hw3_1.cpp
├── hw3_2.cpp
├── hw3_3.cpp
└── hw3_report.pdf
```