2024년 2학기 **운영체제실습** 9주차

CPU Scheduling

System Software Laboratory
School of Computer and Information Engineering
Kwangwoon Univ.

Assignment 3-3

- **CPU scheduling simulator 제작**
  - Write a C program that implements **a simulator using different scheduling algorithms.**
  - The simulator will select tasks from the ready queue based on the scheduling algorithm.
  - There is no need for actual process creation or execution.
  - Print out the task selected to run at each time step in a format similar to a **Gantt chart**.

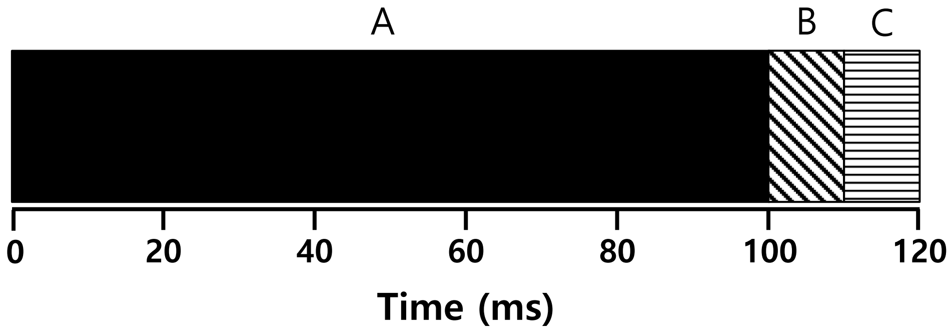


Fig. 1 example of Gantt chart

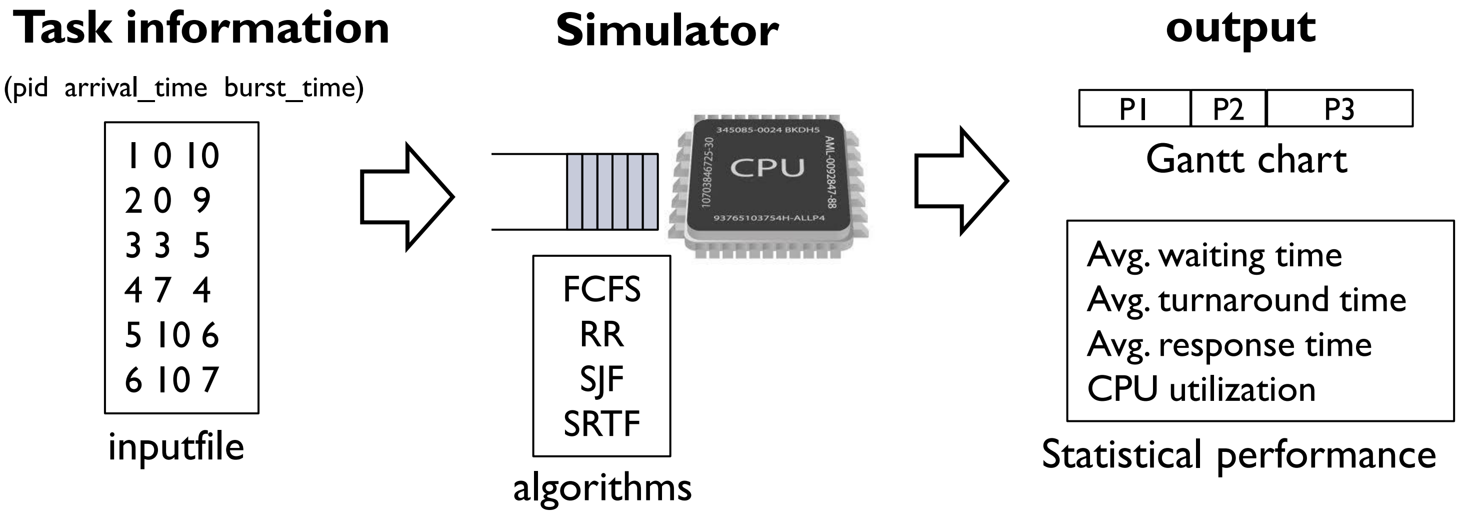
Project Description

- **The scheduling algorithms to be implemented**
  - First Come First Served (**FCFS**)
  - Round Robin (**RR**)
  - Shortest Job First (**SJF**)
  - Shortest Remaining Time First (**SRTF**)
    - also called Shortest Time-to-Completion First (STCF) or Preemptive Shortest Job First (PSJF)

- **FCFS and SJF are non-preemptive, while RR and SRTF are preemptive.**

Project Description

- **CPU scheduler simulator overview**



Task information

(pid arrival_time burst_time)

pid	arrival_time	burst_time
1	0	10
2	0	9
3	3	5
4	7	4
5	10	6
6	10	7

inputfile

Simulator

FCFS
RR
SJF
SRTF

algorithms

output

P1	P2	P3

Gantt chart

Avg. waiting time
Avg. turnaround time
Avg. response time
CPU utilization

Statistical performance

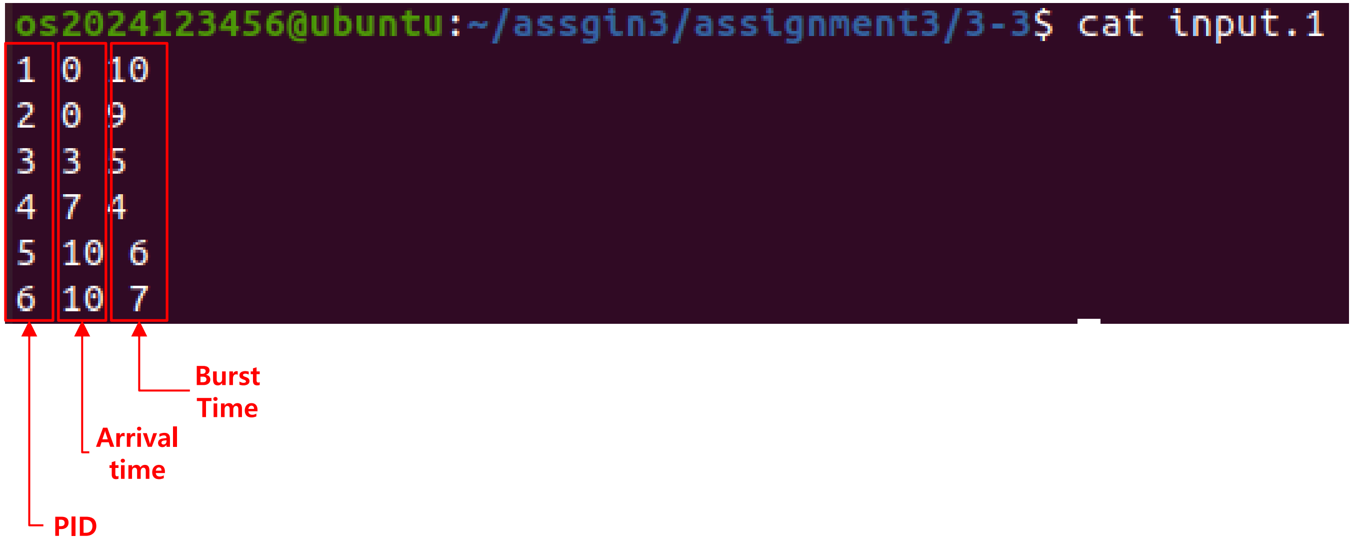
Project requirements

- **1. Implement the scheduling algorithms**
  - FCFS, RR (with time_quantum), SJF, and SRTF.
- **2. The program should print a Gantt chart showing task execution at each time step (in milliseconds).**
  - E.g. | P1 | P1 | P2 | P2 | P3 | P3 | P1 | P1 | P2|
- **3. After all tasks are completed, program must calculate and print**
  - Avg. waiting time, avg. response time, avg. turnaround time, and CPU utilization.
- **4. Context switching overhead is 0.1 milliseconds,**
  - and it should be factored into the simulation.

- **Note: If you use a static array to implement the ready queue, assume the maximum queue length is 1,000.**

Input

- **Task Information**
  - will be read from an input file.
  - The format for each task is:
    - **“pid   arrival_time   burst_time”**
      - **pid** is a unique process ID (integer).
      - **arrival_time** is the time the task arrives (in milliseconds).
      - **burst_time** is the CPU time requested by the task (in milliseconds).
    - Note that all time values are in milliseconds.
    - Input file example (input.1)



```
os2024123456@ubuntu:~/assgin3/assignment3/3-3$ cat input.1
1 0 10
2 0 9
3 3 5
4 7 4
5 10 6
6 10 7
```

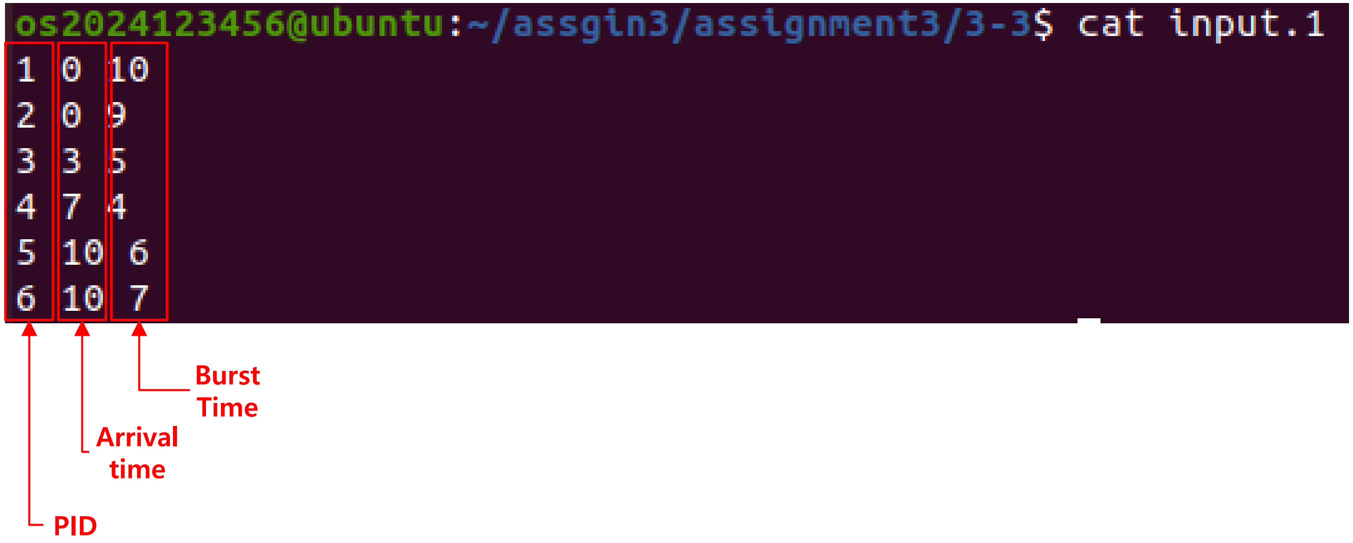
Simulator

- **Command-line Usage**
  - “cpu_simulator input_file [FCFS|RR|SJF|SRTF] [time_quantum]”
    - **input_file** is the file containing task information.
    - The available scheduling algorithms are **FCFS, RR, SJF, and SRTF**.
    - The **time_quantum** parameter is only applicable to the RR algorithm.
      - The time_quantum must be specified for RR.

  - Examples
    - **cpu_simulator input.1 FCFS**
      - Simulate FCFS scheduling using the data file "input.1".
    - **cpu_simulator input.1 RR 2**
      - Simulate RR scheduling with a time quantum of 2 milliseconds using the data file "input.1".
    - **cpu_simulator input.1 SRTF**
      - Simulate Shortest Remaining Time First scheduling using the data file "input.1".

Test Case 01

- **Input file example (input.1)**



```
os2024123456@ubuntu:~/assgin3/assignment3/3-3$ cat input.1
1 0 10
2 0 9
3 3 5
4 7 4
5 10 6
6 10 7
```

Burst Time

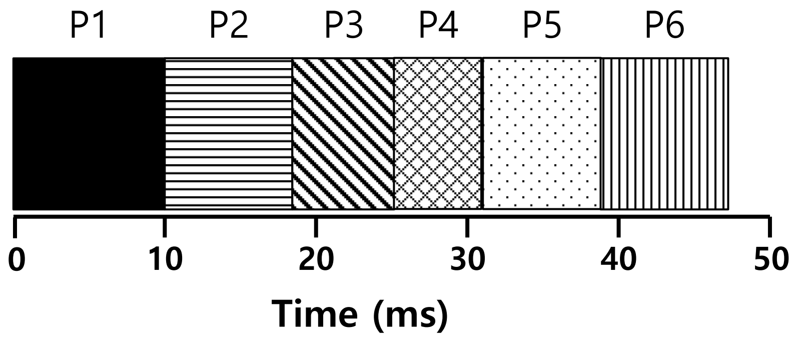
Arrival time

PID

Test Case 01

- **Sample output(FCFS)**

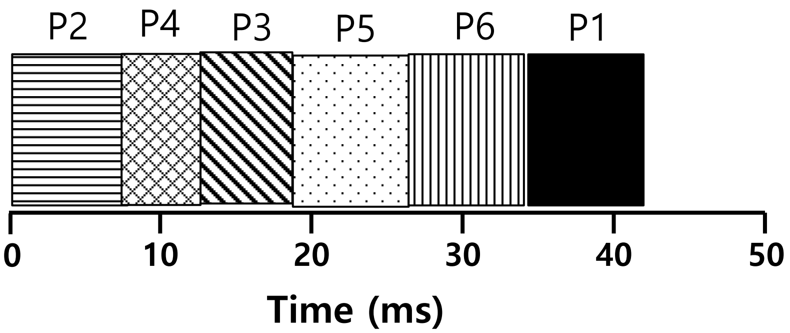
```
os2024123456@ubuntu:~/os/sched$ ./cpu_scheduler   input.1   FCFS
Gantt Chart:
| P1 | P1 | P1 | P1 | P1 | P1 | P1 | P1 | P1 | P1 | P2 | P2 | P2 | P2 | P2 | P2 | P2 | P2 | P2 |P3 | P3 | P3 | P3 | P3 |
 P4 | P4 | P4 | P4 | P5 | P5 | P5 | P5 | P5 | P5 | P6 | P6 | P6 | P6 | P6 | P6 | P6 |
Average Waiting Time = 14.17
Average Turnaround Time = 21.00
Average Response Time = 14.17
CPU Utilization = 98.56%
```



Test Case 01(Cont'd)

- **Sample Output(SJF)**

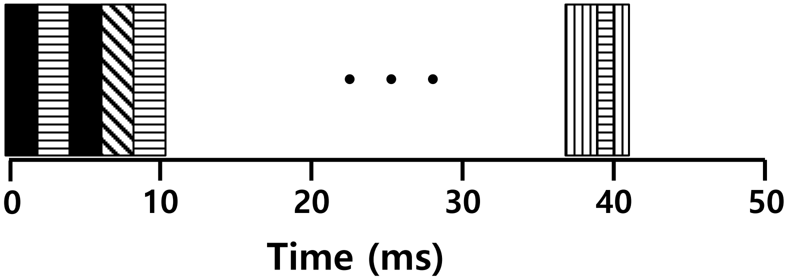
```
os2024123456@ubuntu:~/os/sched$ ./cpu_scheduler   input.1   SJF
Gantt Chart:
| P2 | P2 | P2 | P2 | P2 | P2 | P2 | P2 | P2 | P4 | P4 | P4 | P4 | P3 | P3 | P3 | P3 | P3 | P5 | P5 | P5 | P5 | P5 | P5 |
P6 | P6 | P6 | P6 | P6 | P6 | P6 | P1 | P1 | P1 | P1 | P1 | P1 | P1 | P1 | P1 | P1 |
Average Waiting Time = 10.83
Average Turnaround Time = 17.67
Average Response Time = 10.83
CPU Utilization = 98.56%
```



Test Case 01(Cont'd)

- **Sample output(RR)**

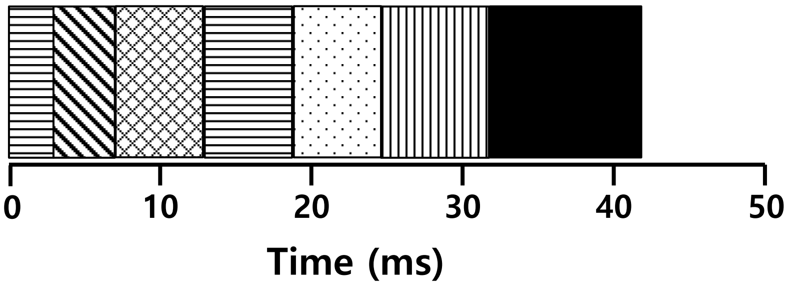
```
os2024123456@ubuntu:~/os/sched$ ./cpu_scheduler   input.1   RR 2
Gantt Chart:
| P1 | P1 | P2 | P2 | P1 | P1 | P3 | P3 | P2 | P2 | P1 | P1 | P4 | P4 | P3 | P3 | P5 | P5 | P6 | P6 | P2 | P2 | P1 | P1 |
 P4 | P4 | P3 | P5 | P5 | P6 | P6 | P2 | P2 | P1 | P1 | P5 | P5 | P6 | P6 | P2 | P6 |
Average Waiting Time = 22.5
Average Turnaround Time = 29.3
Average Response Time = 4.00
CPU Utilization = 94.91%
```



Test Case 01(Cont'd)

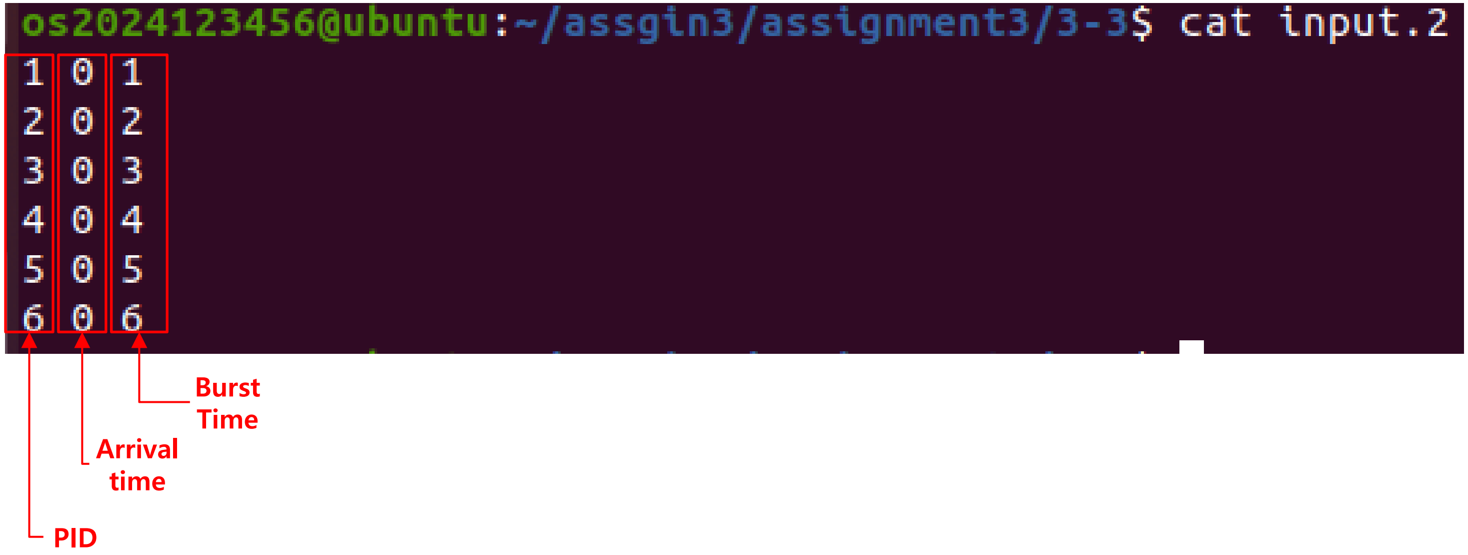
- **Sample Output(SRTF)**

```
os2024123456@ubuntu:~/os/sched$ ./cpu_scheduler   input.1   SRTF
Gantt Chart:
| P2 | P2 | P2 | P3 | P3 | P3 | P3 | P3 | P4 | P4 | P4 | P4 | P2 | P2 | P2 | P2 | P2 | P2 | P5 | P5 | P5 | P5 | P5 | P5 |
 P6 | P6 | P6 | P6 | P6 | P6 | P6 | P1 | P1 | P1 | P1 | P1 | P1 | P1 | P1 | P1 | P1 |
Average Waiting Time = 10.50
Average Turnaround Time = 17.33
Average Response Time = 9.0
CPU Utilization = 98.32%
```



Test Case 02

- **Input file example (input.2)**



```
os2024123456@ubuntu:~/assgin3/assignment3/3-3$ cat input.2
1 0 1
2 0 2
3 0 3
4 0 4
5 0 5
6 0 6
```

Burst Time

Arrival time

PID

Test Case 02(Cont’d)

- **Sample output(FCFS)**

```
os2024123456@ubuntu:~/os/sched$ ./cpu_scheduler   input.2   FCFS
Gantt Chart:
| P1 | P2 | P2 | P3 | P3 | P3 | P4 | P4 | P4 | P4 | P5 | P5 | P5 | P5 | P5 | P6 | P6 | P6 | P6 | P6 | P6 |
Average Waiting Time = 5.83
Average Turnaround Time = 9.3
Average Response Time = 5.83
CPU Utilization = 97.22%
```

- **Sample output(SJF)**

```
os2024123456@ubuntu:~/os/sched$ ./cpu_scheduler   input.2   SJF
Gantt Chart:
| P1 | P2 | P2 | P3 | P3 | P3 | P4 | P4 | P4 | P4 | P5 | P5 | P5 | P5 | P5 | P6 | P6 | P6 | P6 | P6 | P6 |
Average Waiting Time = 5.83
Average Turnaround Time = 9.3
Average Response Time = 5.83
CPU Utilization = 97.22%
```

Test Case 02(Cont'd)

- **Sample output(RR)**

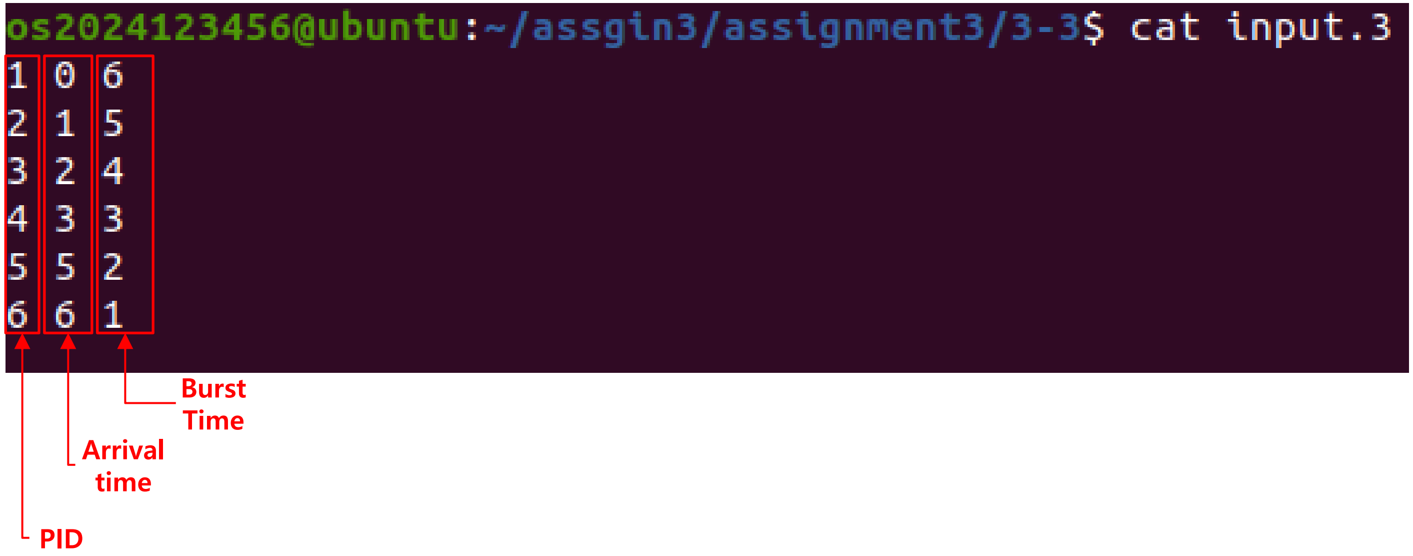
```
os2024123456@ubuntu:~/os/sched$ ./cpu_scheduler   input.2   RR 2
Gantt Chart:
| P1 | P2 | P2 | P3 | P3 | P4 | P4 | P5 | P5 | P6 | P6 | P3 | P4 | P4 | P5 | P5 | P6 | P6 | P5 | P6 | P6 |
Average Waiting Time = 8.17
Average Turnaround Time = 11.67
Average Response Time = 4.17
CPU Utilization = 94.59%
```

- **Sample output(SRTF)**

```
os2024123456@ubuntu:~/os/sched$ ./cpu_scheduler   input.2   SRTF
Gantt Chart:
| P1 | P2 | P2 | P3 | P3 | P3 | P4 | P4 | P4 | P4 | P5 | P5 | P5 | P5 | P5 | P6 | P6 | P6 | P6 | P6 | P6 |
Average Waiting Time = 5.83
Average Turnaround Time = 9.3
Average Response Time = 5.83
CPU Utilization = 97.22%
```

Test Case 03

- **Input file example (input.3)**



```
os2024123456@ubuntu:~/assgin3/assignment3/3-3$ cat input.3
1 0 6
2 1 5
3 2 4
4 3 3
5 5 2
6 6 1
```

Test Case 03(Cont'd)

- **Sample output(FCFS)**

```
os2024123456@ubuntu:~/os/sched$ ./cpu_scheduler   input.3   FCFS
Gantt Chart:
| P1 | P1 | P1 | P1 | P1 | P1 | P2 | P2 | P2 | P2 | P2 | P3 | P3 | P3 | P3 | P4 | P4 | P4 | P5 | P5 | P6 |
Average Waiting Time = 8.83
Average Turnaround Time = 12.3
Average Response Time = 9.0
CPU Utilization = 97.22%
```

- **Sample output(SJF)**

```
os2024123456@ubuntu:~/os/sched$ ./cpu_scheduler   input.3   SJF
Gantt Chart:
| P1 | P1 | P1 | P1 | P1 | P1 | P6 | P5 | P5 | P4 | P4 | P4 | P3 | P3 | P3 | P3 | P2 | P2 | P2 | P2 | P2 |
Average Waiting Time = 5.5
Average Turnaround Time = 9.0
Average Response Time = 5.5
CPU Utilization = 97.22%
```

Test Case 03(Cont’d)

- **Sample output(RR)**

```
os2024123456@ubuntu:~/os/sched$ ./cpu_scheduler   input.3   RR 2
Gantt Chart:
| P1 | P1 | P2 | P2 | P3 | P3 | P1 | P1 | P4 | P4 | P2 | P2 | P5 | P5 | P6 | P3 | P3 | P1 | P1 | P4 | P4 | P2 |
Average Waiting Time = 11.3
Average Turnaround Time = 14.83
Average Response Time = 4
CPU Utilization = 94.59%
```

- **Sample output(SRTF)**

```
os2024123456@ubuntu:~/os/sched$ ./cpu_scheduler   input.3   SRTF
Gantt Chart:
| P1 | P1 | P1 | P1 | P1 | P1 | P6 | P5 | P5 | P4 | P4 | P4 | P3 | P3 | P3 | P3 | P2 | P2 | P2 | P2 | P2 |
Average Waiting Time = 5.83
Average Turnaround Time = 9.3
Average Response Time = 3.83
CPU Utilization = 94.46%
```