

# Lab 6

## FCFS Scheduling



# Objective

- To simulate the **FCFS scheduling**.

# FCFS Scheduling

- ❑ Assigns the CPU based on the order of requests

- ***Nonpreemptive :***

A process keeps running on the CPU until it's blocked or terminated.

✓ **Advantage:** Simple

✗ **Disadvantage:** Short jobs can get stuck behind long jobs

# Quick Refresh

## Turnaround time:

The time of submission to the time of completion.

## Waiting time:

Amount of time a process has been waiting in the ready queue.

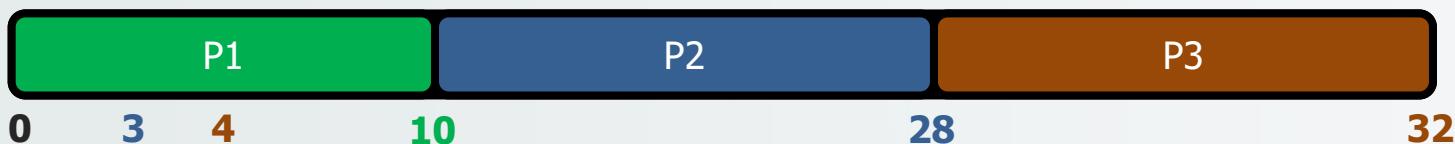
## Response time:

Amount of time it takes from when a request was submitted until the first response is produced.

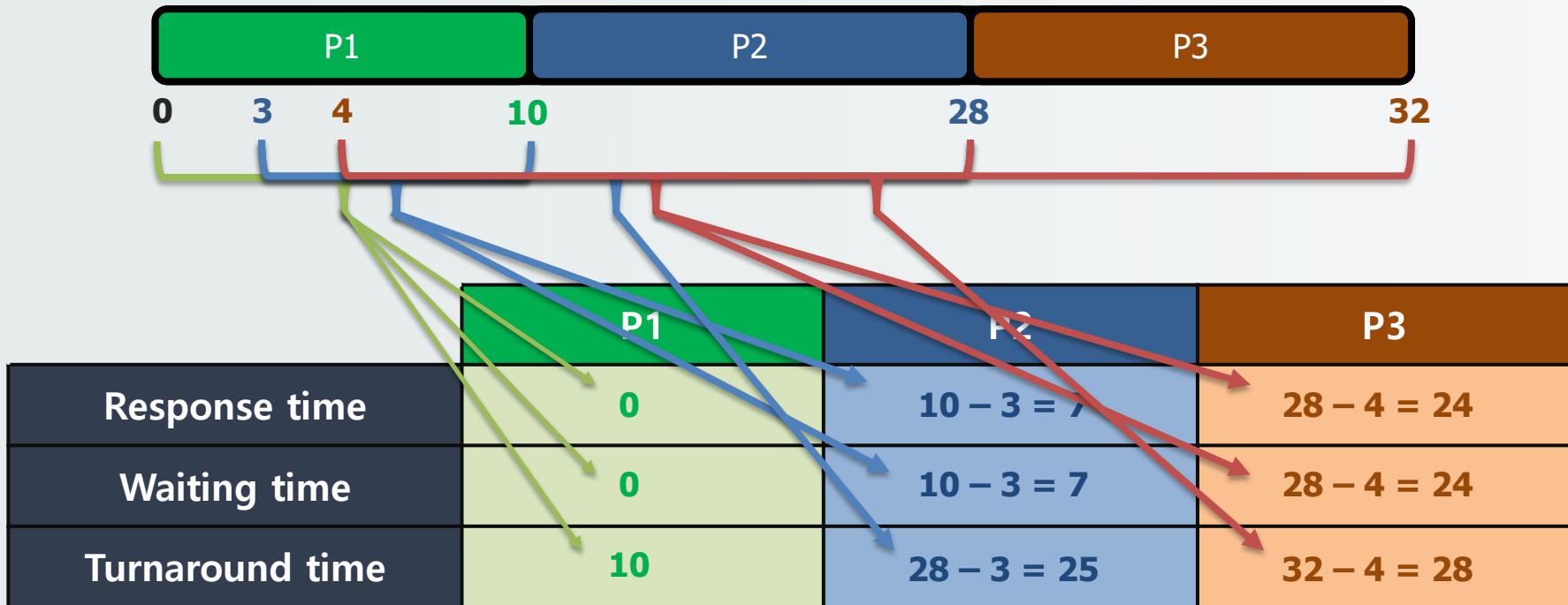


# FCFS Scheduling

Process	Arrive time	Burst time
P1	0	10
P2	3	18
P3	4	4



# FCFS Scheduling

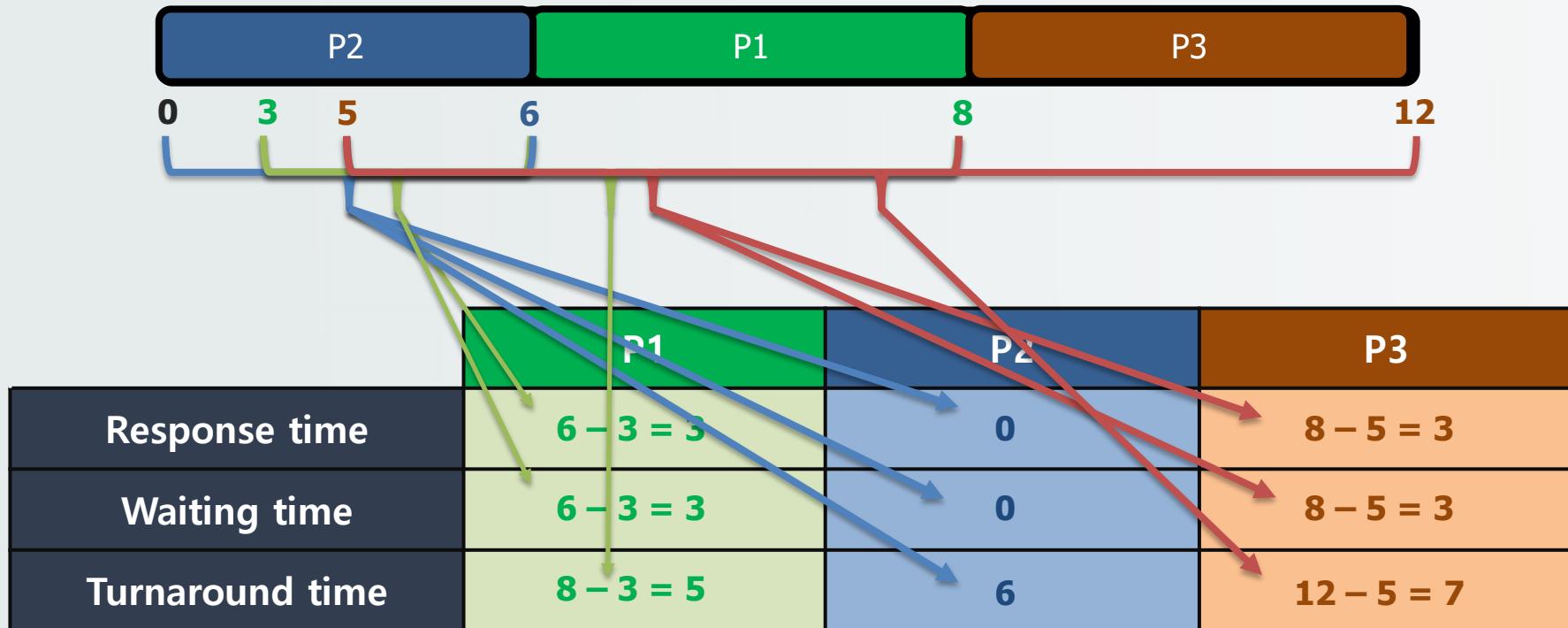


# FCFS Scheduling (unsorted arrival time)

Process	Arrive time	Burst time
P1	3	2
P2	0	6
P3	5	4



# FCFS Scheduling



# Pseudocode

## Calculate Average Wait Time (without arrival time):

wait\_time of first process is 0

For each process (other than the 1<sup>st</sup>) calculate the wait\_time

    wait\_time of current process = wait\_time of previous process + its burst\_time

    Add up Total wait\_time

    average\_wait\_time = Total wait\_time / number of processes

## Calculate Average Turnaround Time (without arrival time):

turnaround\_time of first process is its burst\_time

For each process (other than the 1<sup>st</sup>) calculate the turnaround\_time

    turnaround\_time of current process = turnaround\_time of previous process + burst\_time of current process

    Add up Total turnaround\_time

    average\_turnaround\_time = Total turnaround\_time / number of processes



# FCFS Test Data

Process	Burst Time	Arrival Time	Waiting Time	Turnaround Time
P0	25	0	0	25
P1	4	0	25	29
P2	3	0	29	32
<b>Average</b>			<b>18</b>	<b>28.66</b>
P0	5	0	0	5
P1	8	0	5	13
P2	12	0	13	25
<b>Average</b>			<b>6</b>	<b>14.33</b>
P0	10	0	0	10
P1	18	3	7	25
P2	4	4	24	28
<b>Average</b>			<b>10.3</b>	<b>21.0</b>



# DEMO



# FCFS – Calculate average waiting time

Program fcfs.c

```
1 #include <stdio.h>
2 #define MAX_PROC 3
3
4 struct Process {
5     int burst_t,
6         wait_t;
7 };
8
9 void Calc_AvgWaitTime (struct Process P[]) {
10    float total_wait_t=0.0;
11
12    P[0].wait_t = 0;
13    printf ("P[0] BT: %d WT: %d Total WT: %.2f\n", P[0].burst_t, P[0].wait_t, total_wait_t);
14
15    for (int i = 1; i < MAX_PROC; i++) {
16        P[i].wait_t = P[i-1].wait_t + P[i-1].burst_t;
17        total_wait_t += P[i].wait_t;
18
19        printf ("P[%d] BT: %d WT: %d Total WT: %.2f\n", i, P[i].burst_t, P[i].wait_t, total_wait_t);
20    }
21
22    printf ("Average WT: %.2f \n", total_wait_t/MAX_PROC);
23}
24
25 int main () {
26    struct Process P[MAX_PROC] = {{25,0}, {4,0}, {3,0}};
27
28    Calc_AvgWaitTime (P);
29    return 0;
30}
```

```
~/Lab6$ gcc fcfs.c
~/Lab6$ ./a.out
P[0] BT: 5 WT: 0 Total WT: 0.00
P[1] BT: 8 WT: 5 Total WT: 5.00
P[2] BT: 12 WT: 13 Total WT: 18.00
Average WT: 6.00
~/Lab6$ 
~/Lab6$ gcc fcfs.c
~/Lab6$ ./a.out
P[0] BT: 25 WT: 0 Total WT: 0.00
P[1] BT: 4 WT: 25 Total WT: 25.00
P[2] BT: 3 WT: 29 Total WT: 54.00
Average WT: 18.00
~/Lab6$
```

# Resources

1. CPU Scheduling

<https://www.youtube.com/watch?v=EWkQI0n0w5M>

2. Scheduling Algorithms - First Come First Served (FCFS)

<https://www.youtube.com/watch?v=7DoP1L9nAAs>

3. Scheduling Criteria

<https://www.youtube.com/watch?v=bWHFY8-rL5I>

