# CSE3241: Operating System and System Programming

Class-15


Sangeeta Biswas, Ph.D.

Assistant Professor
Dept. of Computer Science and Engineering (CSE)
Faculty of Engineering
University of Rajshahi (RU)
Rajshahi-6205, Bangladesh
E-mail: sangeeta.cse@ru.ac.bd / sangeeta.cse.ru@gmail.com

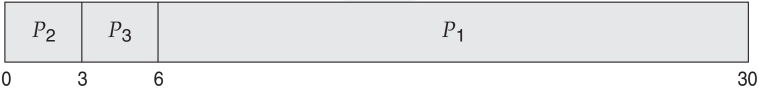

November 26, 2017

# CPU Scheduling Algorithms: FCFS

- In**First-Come, First-Served (FCFS)** scheduling, process **P** that **requests the CPU first** is allocated the CPU first.

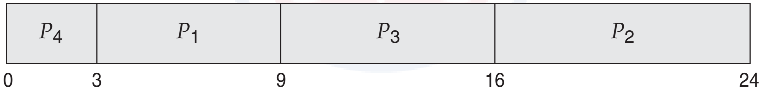
- The average waiting time (AWT) is often quite long and highly dependent on the order of process arrival.

- Say $P_1$, $P_2$ and $P_3$ arrive at time 0 demanding for CPU burst of 24, 3 and 3 miliseconds(ms) respectively. 1st case AWT: 17ms and 2nd case AWT: 3ms.

# CPU Scheduling Algorithms: SJF

- In **Shortest-Job-First (FJS)** scheduling, process **P** that requires the **least CPU access** is allocated the CPU first.
- The average waiting time is optimal for a given set of processes.
- Knowing the CPU usage time in advance is difficult.
- Say $P_1$, $P_2$, $P_3$ and $P_4$ arrive at time 0 demanding for CPU burst of 6, 8, 7 and 3 ms respectively. AWT: 7ms whereas in FCFS AWT: 10.25ms.

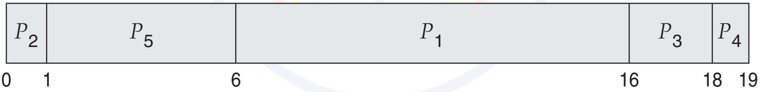


# CPU Scheduling Algorithms: PB

- In **Priority based (PB)** scheduling, process **P** with the **highest priority** is allocated the CPU first.

- Equal priority processes are scheduled in FCFS order.

- Say $P_1$, $P_2$, $P_3$, $P_4$ and $P_5$ arrive at time 0 demanding for CPU burst of 10, 1, 2, 1 and 5 ms with priority $\{3, 1, 4, 5, 2\}$, respectively. AWT: 8.2ms.



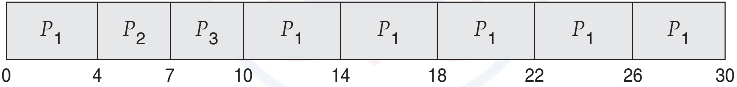

- Major problem is indefinite blocking or starvation.

# CPU Scheduling Algorithms: RR

- **Round-Robin (RR)**: Process **P** is allocated the CPU for **T** time and after that it is put in the **last of the ready queue**.

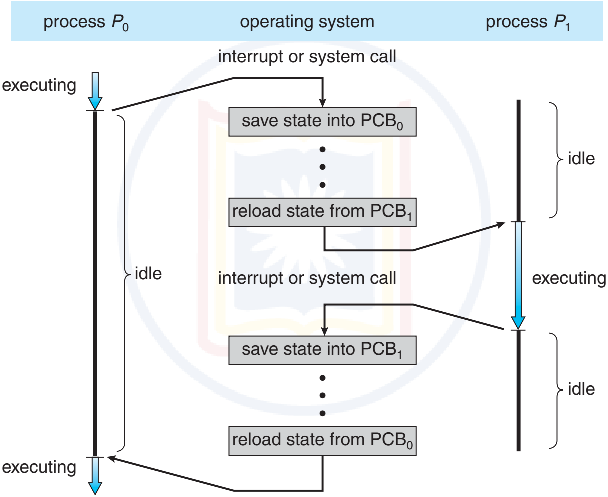
- Average waiting time of processes is often long.

- Say $P_1$, $P_2$, and $P_3$ arrive at time 0 demanding for CPU burst of 24, 3, and 3, respectively. AWT: 5.66ms.




- Has huge effect of context switching.

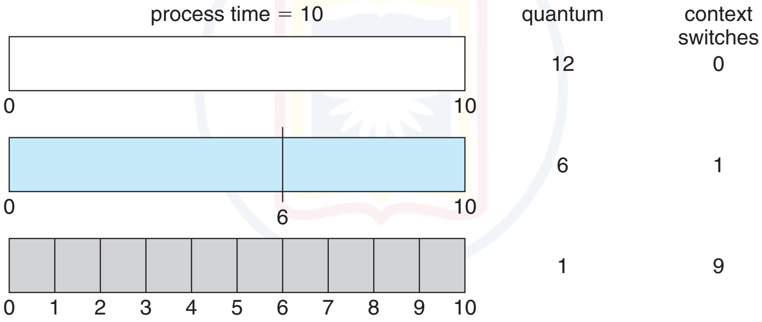
# Context Switching [?]



# Context Switches in Round Robin Scheduling [?]

- Let we have only one process $P$ of 10 time units.

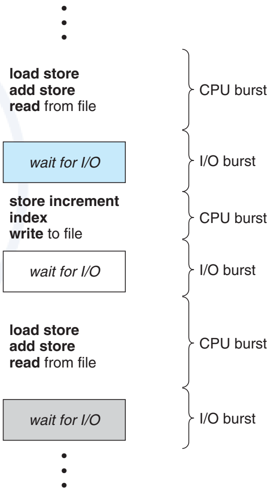
Figure: How a smaller time quantum increases context swithes



# CPU bound Process vs. I/O bound Process [?]

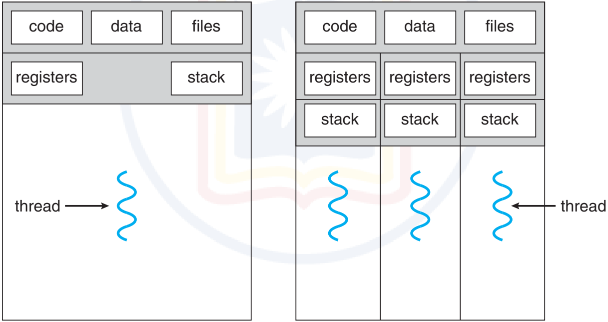
- Processes execution consists of a cycle of CPU execution and I/O wait.
- A CPU-bound process:
  - generates I/O requests infrequently, using more of its time doing computations.
  - might have a few long CPU-bursts.
- An I/O bound process:
  - is one that spends more of its time doing I/O than spends doing computations
  - has many short CPU bursts.

Sequences of CPU and I/O bursts



# Single-threaded vs. Multi-threaded Processes [?]

- A thread is a basic unit of programmed instructuctions that can be managed independently by a CPU scheduler.
- We generally write programs which run as single-threaded processes.
- Threads belong to the same process run concurrently.




single-threaded process multithreaded process

## Single-Threaded Program

```
#include<stdio.h>

void sum(){
    while(1){
        printf("5 + 3: %d.", 5 + 3 );
    }
}

int main(){

    sum();

    while(1){
        printf("I am unstoppable.");
    }

    return 0;
}
```

# pthread_create()

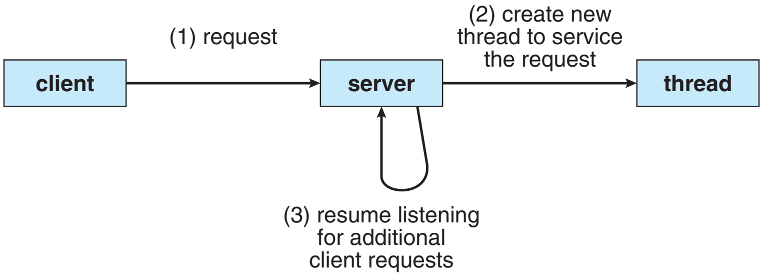
- Purpose: To create a new thread.
- Header File: #include <pthread.h>
- Synopsis:

```
int pthread_create(
    pthread_t *thread,
    const pthread_attr_t *attr,
    void *(*start_routine) (void *),
    void *arg
);
int pthread_create(
    pthread_t *thread,
    NULL,
    void *(*start_routine) (void *),
    NULL
);
```

- Return Value: **0** on success and a **error number** on failure.

# Multi-Threaded Server Architecture [?]

- Your next target is to turn TCPserver process into multi-threaded process.



# References

P. B. Galvin A. Silbeschatz and G. Gagne.
*Operating System Concepts*.
John Wiley & Sons, 9 edition, 2012.

Wikipedia.