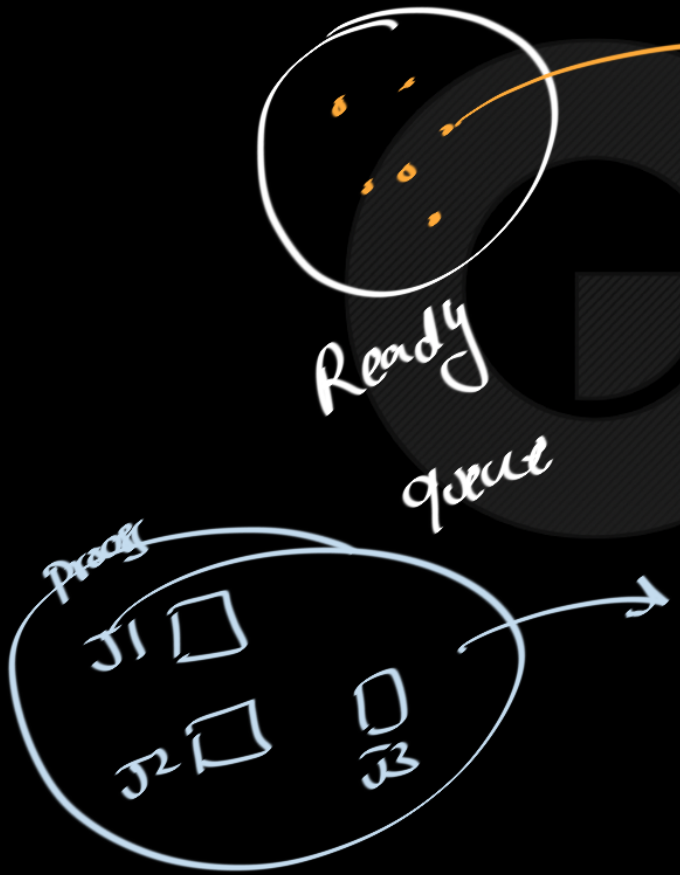
Lecture 9



- max CPU utilisation
- max throughput
- minimise avg Response time
- minimise avg waiting time

Scheduling Criteria

- **CPU utilization** – keep the CPU as busy as possible
- **Throughput** – # of processes that complete their execution per time unit
- **Turnaround time** – amount of time to execute a particular process
- **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, not output (for time-sharing environment)

Scheduling Criteria

- **CPU utilization** – keep the CPU as busy as possible max.
- **Throughput** – # of processes that complete their execution per time unit max
- **Turnaround time** – amount of time to execute a particular process min
- **Waiting time** – amount of time a process has been waiting in the ready queue min
- **Response time** – amount of time it takes from when a request was submitted until the first response is produced, not output (for time-sharing environment) min

6.2 Scheduling Criteria

Different CPU-scheduling algorithms have different properties, and the choice of a particular algorithm may favor one class of processes over another. In choosing which algorithm to use in a particular situation, we must consider the properties of the various algorithms.

Many criteria have been suggested for comparing CPU-scheduling algorithms. Which characteristics are used for comparison can make a substantial difference in which algorithm is judged to be best. The criteria include the following:

- **CPU utilization**. We want to keep the CPU as busy as possible. Conceptually, CPU utilization can range from 0 to 100 percent. In a real system, it should range from 40 percent (for a lightly loaded system) to 90 percent (for a heavily loaded system).
- **Throughput**. If the CPU is busy executing processes, then work is being done. One measure of work is the number of processes that are completed per time unit, called **throughput**. For long processes, this rate may be one process per hour; for short transactions, it may be ten processes per second.
- **Turnaround time**. From the point of view of a particular process, the important criterion is how long it takes to execute that process. The interval from the time of submission of a process to the time of completion is the turnaround time. Turnaround time is the sum of the periods spent waiting to get into memory, waiting in the ready queue, executing on the CPU, and doing I/O.
- **Waiting time**. The CPU-scheduling algorithm does not affect the amount of time during which a process executes or does I/O. It affects only the amount of time that a process spends waiting in the ready queue. Waiting time is the sum of the periods spent waiting in the ready queue.

Optional Read

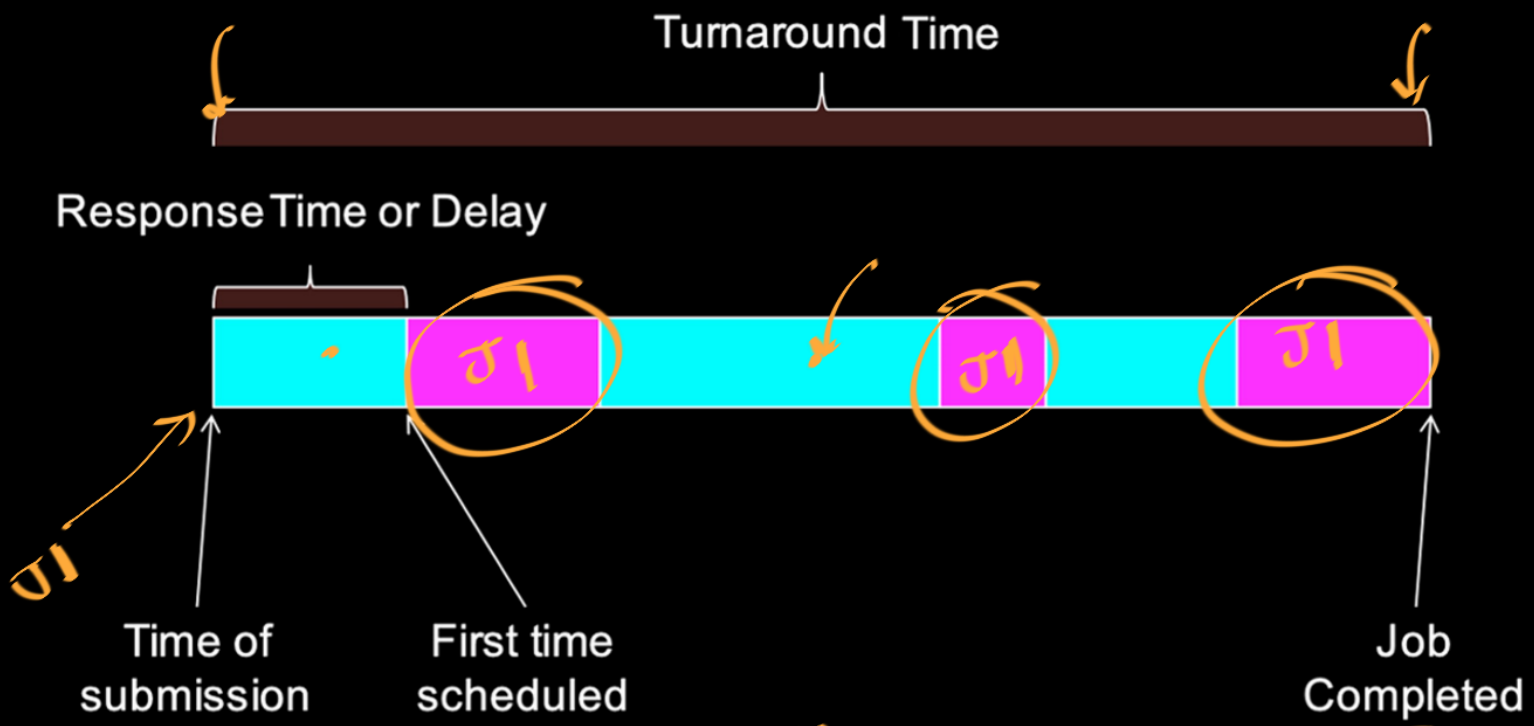
Source: Galvin

- **Response time**. In an interactive system, turnaround time may not be the best criterion. Often, a process can produce some output fairly early and can continue computing new results while previous results are being output to the user. Thus, another measure is the time from the submission of a request until the first response is produced. This measure, called response time, is the time it takes to start responding, not the time it takes to output the response. The turnaround time is generally limited by the speed of the output device.

GO CLASSES

Understanding

- Turn around time
- waiting time
- Response time



Waiting Time: sum of "▭" periods (time on ready queue but not running)

http://www.cs.cornell.edu/courses/cs4410/2015fa/slides/05-scheduling.pdf

GO CLASSES

Arrival time = 0
A, B, C

Response time of A = 0

Waiting time of A = 5

Turn around time of A = 9

GO CLASSES

Arrival time = 0

A, B, C

Response time of B =

Waiting time of B =

Turn around time of B =

GO CLASSES

Arrival time = 0
A, B, C

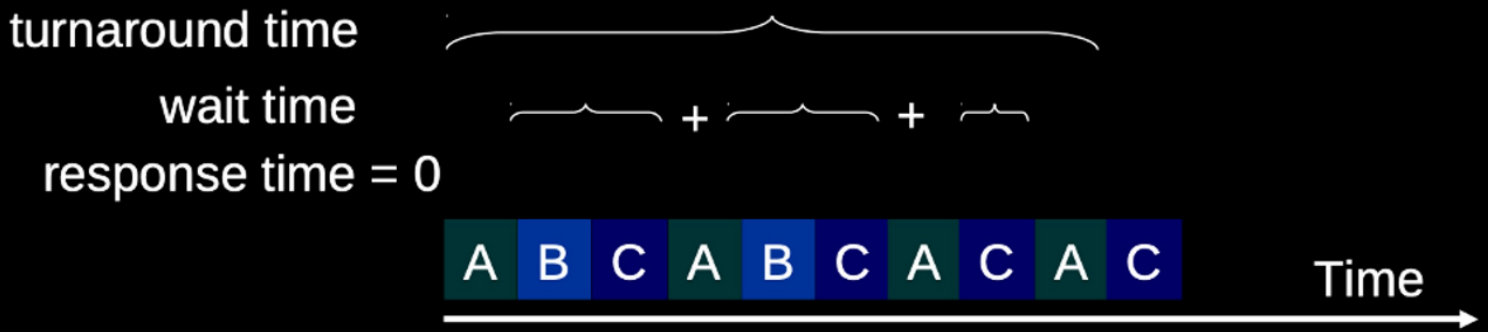
H.W.

Response time of C =

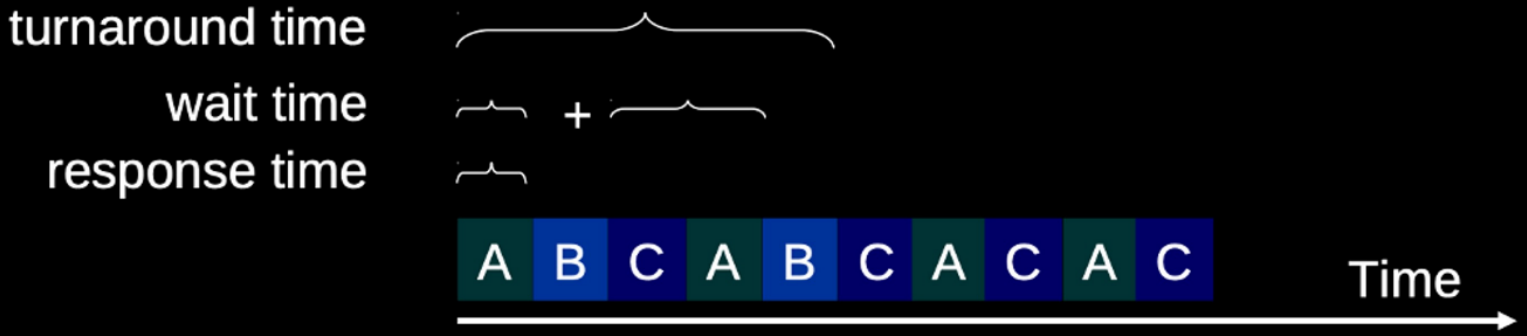
Waiting time of C =

Turn around time of C =

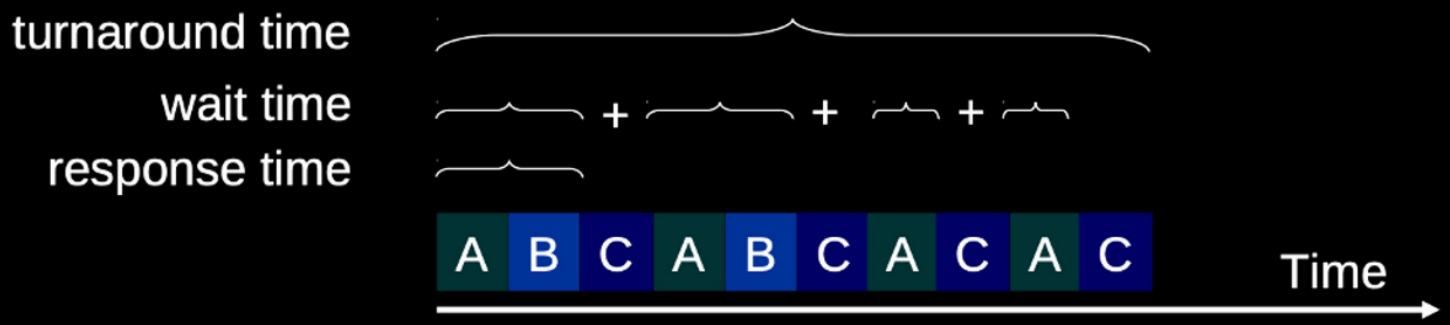
- Suppose we have processes A, B, and C, submitted at time 0.
- We want to know the response time, waiting time, and turnaround time of process A



- Suppose we have processes A, B, and C, submitted at time 0.
- We want to know the response time, waiting time, and turnaround time of process B.



- Suppose we have processes A, B, and C, submitted at time 0.
- We want to know the response time, waiting time, and turnaround time of process C



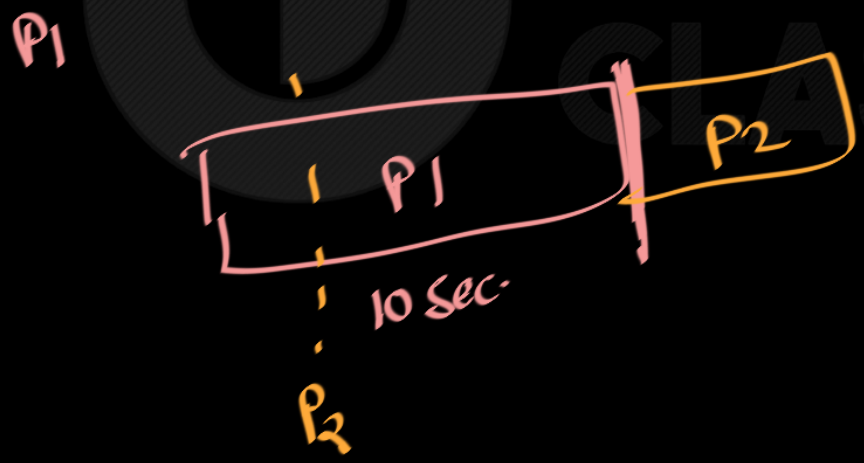
GO CLASSES

Now we know what are the metrics to look for (when designing scheduling algorithms)

Scheduling Algorithms

First-Come First-Serve (FCFS) (FIFO)

- By far the simplest CPU-scheduling algorithm



First-Come First-Serve (FCFS)

- By far the simplest CPU-scheduling algorithm

Process	Burst Time
P_1	24
P_2	3
P_3	3

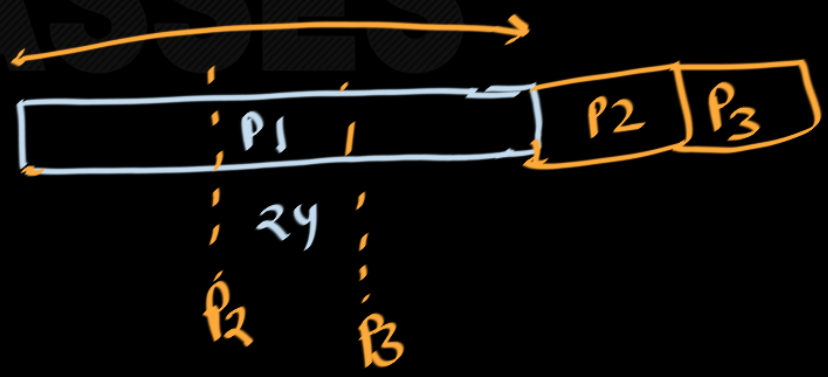
CPU

Arrival time

1

10

15



First-Come First-Serve (FCFS)

- By far the simplest CPU-scheduling algorithm

Process	Burst Time
P_1	24
P_2	3
P_3	3

Arrival time = 0 for all Processes

Response time of P_1 = 0

Waiting time of P_1 = 0

Turn around time of P_1 = 24

Response time of P_2 = 24

Waiting time of P_2 = 24

Turn around time of P_2 = 27

Response time of P_3 = 27

Waiting time of P_3 = 27

Turn around time of P_3 = 30

Response time of P_1 = 0

Waiting time of P_1 = 0

Turn around time of P_1 = 24

Response time of P_2 = 24

Waiting time of P_2 = 24

Turn around time of P_2 = 27

Response time of P_3 = 27

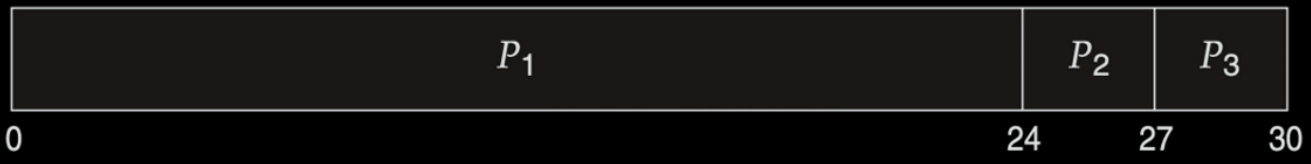
Waiting time of P_3 = 27

Turn around time of P_3 = 30

Avg waiting time (Avg Response time) = $\frac{0+24+27}{3}$

Avg Turn around time $\frac{24+27+30}{3}$

If the processes arrive in the order P_1, P_2, P_3, and are served in FCFS order, we get the result shown in the following **Gantt chart**, which is a bar chart that illustrates a particular schedule, including the start and finish times of each of the participating processes:

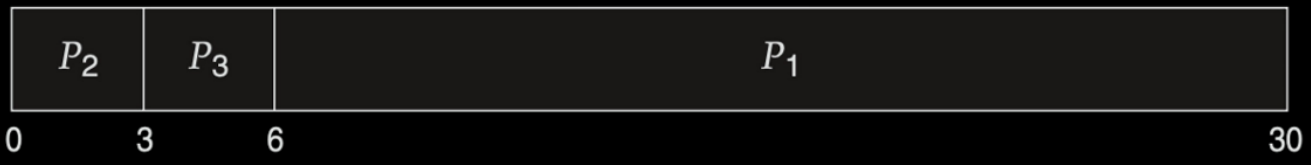


$P_1 \rightarrow P_2 \rightarrow P_3$

Arrival times

Avg. waiting time = 17

The waiting time is 0 milliseconds for process P_1, 24 milliseconds for process P_2, and 27 milliseconds for process P_3. Thus, the average waiting time is $(0 + 24 + 27)/3 = 17$ milliseconds. If the processes arrive in the order P_2, P_3, P_1, however, the results will be as shown in the following Gantt chart:



$P_2 \rightarrow P_3 \rightarrow P_1$

Arrival times

The average waiting time is now $(6 + 0 + 3)/3 = 3$ milliseconds. This reduction is substantial. Thus, the average waiting time under an FCFS policy is generally not minimal and may vary substantially if the processes' CPU burst times vary greatly.

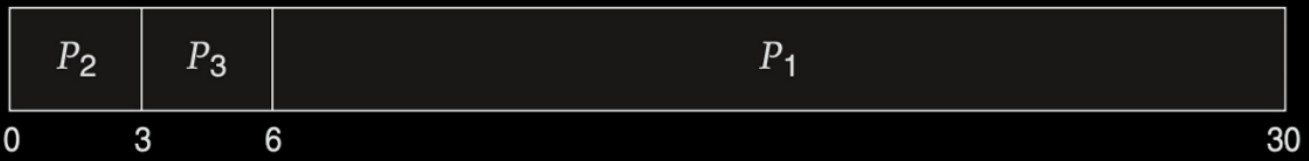
Source: Galvin

Arg waiting time = 17 ms

$P_1 \rightarrow P_2 \rightarrow P_3$

Arrival times

Avg. waiting time = 17



Arg waiting time = 3 ms

$P_2 \rightarrow P_3 \rightarrow P_1$

Arrival times

Source: Galvin

Convoy Effect

because of large Process Scheduled first, other Process has to Suffer.

GATE 23

https://compas.cs.stonybrook.edu/~nhonarmand/courses/fa17/cse306/slides/09-sched.pdf