

---

# **Process Scheduling**

JAC Hermocilla

2023-08-26

## Process Scheduling

### Learning Outcomes

At the end of this session, the students should be able to:

1. describe the different process scheduling algorithms; and
2. implement simulations of the different process scheduling algorithms.

### Software Requirements

1. Ubuntu 20.04 or higher with build tools

### Process Scheduling

A process usually spends its run time in either executing instructions (CPU burst) or waiting for I/O completion (I/O burst). Assuming we have a single CPU, when waiting for I/O, the CPU is idle, doing nothing. In multiprogramming and time-sharing systems, several processes are allowed to run to maximize CPU utilization. When a process is waiting for I/O completion, another process is allowed to execute, keeping the CPU busy. Since CPU is one of the primary computer resources, scheduling of this kind is an important function to the operating system.

Selecting the process to execute is the task of the process scheduler module of an operating system. The process scheduler has access to the set of processes that are in the READY state. This set is often referred to as the ready queue. Processes in the ready queue are often referred to as blocked processes. Given the important role of the process scheduler, it is frequently invoked, and thus, hooked to the timer interrupt. The dispatcher is another process scheduling component which gives control of the CPU to the process selected by the scheduler.

### Preemptive and Nonpreemptive Scheduling

Process scheduling decisions take place when a process (1) switches from running to waiting state, (2) switches from running to ready state, (3) switches from waiting to ready state, or (4) terminates. These instances can be preemptive or nonpreemptive. In nonpreemptive systems, once the process scheduler has selected a process to execute, the process is allowed to run until completion. This happens when a new process must be selected, for instance, the first and fourth situations stated above. In preemptive systems, a process may be removed from the CPU when a new process arrives, even if it is not yet done with its task. This happens when certain conditions for preemption are met, for instance, the second and third situations described above. Modern operating systems are preemptive.

## Scheduling Criteria

There are several algorithms for process scheduling. To determine which scheduling algorithm is best for a particular situation, certain criteria must be considered:

- *CPU Utilization*. The percentage of time a process spends in the CPU relative to all the other processes.
- *Throughput*. The total number of processes completed per unit time.
- *Turnaround Time*. The total amount of time used up by a process (starting from its arrival in the ready queue up to its completion).
- *Waiting Time*. The total amount of time a process spent waiting in the ready queue
- *Response Time*. The amount of time it took for the process to wait in the ready queue until its first allocation in the CPU (from its arrival in the ready queue until its first execution).