



OPERATING SYSTEMS COURSE

Jesús Carretero Pérez
David Expósito Singh
José Daniel García Sánchez
Francisco Javier García Blas
Florin Isaila

Topic 3 Presentation Guide: Concurrency and Synchronization.

This topic presents problems and solutions related to communication and synchronization of processes running concurrently on a computer. Processes running concurrently in a system can be classified as independent or cooperating processes. An independent process is one that runs without requiring help or cooperation from other processes. A clear example of independent processes are different command interpreters that run simultaneously on a system. Processes are cooperating when they are designed to work together in an activity, so they must be able to communicate and interact with each other.

Whether the processes are independent or cooperating, a series of interactions between them can occur. These interactions can be of two types:

- Interactions motivated because processes share or compete for access to physical or logical resources. This situation arises in the different types of processes discussed above. For example, two completely separate processes can compete for disk access. In this case the operating system must arrange for the two processes to access orderly the resource without conflict. This situation also appears when multiple processes want to change the content of a record in a database. Here is the manager of the database which will have to take care of ordering the various accesses to the registry.
- Interactions motivated because processes communicate and synchronize with each other to achieve a common goal. For example, a compiler can be constructed by two processes: the compiler itself, which is responsible for generating assembly code, and the linker, process to obtain machine language code from the assembly. In this example the need to communicate and synchronize the two processes it can be seen.

In a computer system concurrently multiple processes run and they are usually not independent, but competing in the exclusive use of resources and communicate and synchronize with each other. The operating system must take care to ensure that such interactions are conducted properly by providing mechanisms and services that enable communication and synchronization between processes to run concurrently, and the mutual exclusion required by them.

The topic includes three lessons:

- Concurrency and Communication and Synchronization problems.
- Threads Communication and Synchronization.
- Development of current servers.

The primary objective of the subject is that students understand the fundamental problem with the execution of concurrent processes cooperating with each other, and the need to communicate and synchronize these processes correctly

This general objective is broken down into a series of specific objectives, which are listed below:

- To introduce the concept of concurrency and advantages and problems associated with it, such as race conditions .
- To present the concepts of mutual exclusion and critical section.
- To show students the mechanisms of communication and synchronization of processes and threads: semaphores as primary synchronization mechanism for processes and threads.
- To present mutex and condition variables for use in threads.
- To show and solve classic concurrency issues, such as the producer-consumer problem and the problem of writers-readers.
- To teach students to design and build servers based on concurrent processes and threads .

Associated Materials

The theoretical lessons are complemented with exercises for each of them, to be resolved after the classes, as specified in the schedule associated with the teaching guide.

Mandatory and recommended reading with more problems and solved exercises are included, so that students can complete their training.

This topic is complemented with a lab to use systems calls for concurrency.