

King Saud University
Department of Computer Science
CSC227: Operating Systems
Tutorial – Chapter 6: Process Synchronization

Q 1) What do we mean by process synchronization?

Process **Synchronization** means sharing system resources by processes in a such a way that, Concurrent access to **shared data** is handled thereby minimizing the chance of **inconsistent data**. Maintaining data consistency demands mechanisms to ensure synchronized execution of cooperating processes.

Q 2) What is a critical section?

A critical section (CS) is a sequence of instructions that can be executed by **at most one** process at the same time.

Q 3) What is a critical section problem?

Critical section problem is to design protocol to solve **concurrent access** to **shared data** among **various processes**.

Q 4) List the requirements that a solution to the critical section problem must satisfy.

- **Mutual Exclusion** - If process P_i is executing in its critical section, then no other processes can be executing in their critical sections
- **Progress** - If no process is executing in its critical section and there exist some processes that wish to enter their critical section, then the selection of the processes that will enter the critical section next cannot be postponed indefinitely
- **Bounded Waiting** - A bound must exist on the number of times that other processes are allowed to enter their critical sections after a process has made a request to enter its critical section and before that request is granted

Q 5) What is the meaning of the term busy waiting? What other kinds of waiting are there in an operating system? Can busy waiting be avoided altogether? Explain your answer.

Busy waiting means that a process is **waiting** for a condition to be satisfied in a **tight loop** without relinquishing the processor. Alternatively, a process could wait by relinquishing the processor, and block on a condition and wait to be awakened at some appropriate time in the future. Busy waiting can be avoided but incurs the overhead associated with putting a process to sleep and having to wake it up when the appropriate program state is reached.

Q 6) Which process can be affected by other processes executing in the system?

Cooperating process

Q 7) What is race condition?

When several processes access the same data concurrently and the outcome of the execution depends on the particular order in which the access takes place, is called race condition.

Q 8) Which one of the following is a synchronization tool?

Semaphore

Q 9) When high priority task is indirectly preempted by medium priority task effectively inverting the relative priority of the two tasks, the scenario is called: priority inversion.

Q 10) Explain why spinlocks are not appropriate for single-processor systems yet are often used in multiprocessor systems.

Spinlocks are not appropriate for single-processor systems because the condition that would break a process out of the spinlock can be obtained only by executing a different process. If the process is not relinquishing the processor, other processes do not get the opportunity to set the program condition required for the first process to make progress.

In a multiprocessor system, other processes execute on other processors and thereby modify the program state in order to release the first process from the spinlock.