(1) How does “SRTF” process scheduling algorithm work?

SRTF (Shortest Remaining Timing First) is a process scheduling algorithm, which assigns a processor to the process that has the shortest execution time after the current time. It is a preemptive version of SFJ (a processor is switched to another process when the current process finishes or if a new process that has shorter remaining execution time is submitted for execution (the current process will be preempted)).

(2) What is “non-preemptive process scheduling”?

The non-preemptive process scheduling is a type of process scheduling algorithms, in which once a processor is assigned to a process, the processor will not be taken away from the process unless (a) the process finishes running or (b) the process voluntarily releases the processor (e.g., for performing I/O requests, such as waiting for inputs from the keyboard).
(3) How can “race condition” happen? Show “how” using an example.

The race condition is the situation where the outputs from processes are never predictable (i.e., each time the process is executed, its outputs can be different), because of unpredictable timing of the process to be preempted (or “unpredictable context-switching”) by another process.

Note: examples should be using assembly instructions (without assembly instructions, race condition will not be explained).

(4) What is “critical section”?

Any one of the followings:

- The term “critical section” means a group of instructions where a process can cause race condition.
- The term “critical section” means a group of instructions where at most one process can execute at a time.

Note: the following solution is incorrect:

- The term “critical section” means a group of instructions where a process should not be interrupted.
(5) What “signal” system call to a semaphore exactly performs?

Signa

If no one waiting on S, set S = 1
If some one waiting on S, let the first proceed to CS and leave S = 0