What is “preemptive process scheduling”?

Preemptive process scheduling is a type of process scheduling (a decision making process for the OS to select the next process to be assigned a processor) in such a way that the operating system can take away a processor from the process that is currently assigned the processor even if the process does not want to release the processor (and the operating system can assign the processor to another process in the ready state).

Note: the underlined concept should be emphasized for full credit.

How does “RR” process scheduling algorithm work?

The RR (Round-Robin) process scheduling algorithm is a preemptive process scheduling algorithm that assigns a processor to each process in the ready state only for a set amount of time (“processor time slice”). At the end of each processor time slice, the operating system preempts the currently running process, takes away the processor assigned to it, and assigns it another process in ready state (so that every process in the ready state will be assigned a processor before a process is assigned a processor for its second time).

What is “turnaround time” (in the context of process scheduling)?

The term, “turnaround time”, means the time (a) when a process is requested for an execution (not “when a process starts running”) (b) until one finishes running.

Note: for full credit both (a) and (b) should be presented.
(4) Which process scheduling algorithms can cause “process starvation” (select all that apply)?

(a) FCFS
(b) RR
(c) SJF
(d) SRTF

**Solutions:** (c) and (d)

**Grading Policies:**

- For selecting (c): +5 points
- For selecting (d): +5 points
- For selecting any other option(s): -5 points
- If the total score of this question is negative, 0 point for this question

(5) How can “race condition” happen? Show “how” using an example.

**Solution:** When two processes (P₁ and P₂) try to update a variable in shared memory (as shown above), and if the processor is contexed-switched (its timing is unpredictable) after P₁ finished “ADD” instruction. If the processor is assigned to P₂ after that, which finishes its all three instructions in the example, the content of ‘A’ will be ‘8’ (instead of ‘5’) when P₁ finishes its third instruction (assuming the a processor is re-assigned to P₁).

**Note:** Examples should use assembly instructions (i.e., any example that does not use assembly instructions will not be able to explain how race condition can happen).