List of the Possible Questions

#1: What do “sufficient conditions” guarantee?

#2: What do “necessary conditions” guarantee?

#3: If a sufficient condition is not satisfied, what conclusion can we draw?

#4: If a necessary condition is satisfied, what conclusion can we draw?

#5: How does “FCFS” process scheduling algorithm work?

#6: How does “RR” process scheduling algorithm work?

#7: How does “SJF” process scheduling algorithm work?

#8: How does “SRTF” process scheduling algorithm work?

#9: What is the “preemptive process scheduling”? 

#10: What is the “non-preemptive process scheduling”?

#11: What is “throughput” (in the context of process scheduling)?

#12: What is “response time” (in the context of process scheduling)?

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

#14: What is “process starvation”?

#15: Which process scheduling algorithms can cause “process starvation” (select all that apply)?

   (1) FCFS
   (2) RR
   (3) SJF
   (4) SRTF

#16: What is “race condition”?
#17: How can “race condition” happen? Show “how” using an example.

#18: What is “critical section”?  

#19: What is “mutual exclusion”?  

#20: What does “atomic” in “atomic operations” mean?  

#21: What is “a binary (or mutex) semaphore”?  

#22: What are the two primary system calls for manipulating a semaphore?  

#23: Explain how a semaphore can prevent race condition.  

#24: What are the two operations of a semaphore (just name them)?  

#25: What “wait” system call to a semaphore exactly performs?  

#26: What “signal” system call to a semaphore exactly performs?  

#27: Why must the two system calls for semaphores (“wait” and “signal”) be atomic operations (explain the reason)?  

#28: What are “counting semaphores” (how are “counting semaphores” different from “binary (mutex) semaphores”?  

#29: Who manages semaphores?  

#30: Operating systems use “queue (FIFO data structure)” for managing processes blocked on a semaphore. Why is FIFO-queue used (the best reason for using FIFO structure)?