List of the possible questions

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

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

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

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

#5: What is “preemptive process scheduling”?

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

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

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

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

#10: What are the two processor modes operating systems set?

#11: What is “kernel (system) mode” of a processor?

#12: What are “system calls”?

#13: What is “user mode” of a processor?

#14: Why do operating systems have two processor modes (what problems will happen if there is no “user mode” – all programs run in the kernel (system) mode)?

#15: What is “race condition”?

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

#17: What is “critical section”?

#18: What is “mutual exclusion”?

#19: What does “atomic” in “atomic operations” mean?
#20: What is “a binary (or mutex) semaphore”? 

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

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

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

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

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

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

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

#28: Who manages semaphores? 

#29: 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)?