List of the possible questions

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

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

#3: What are “system calls”?

#4: What is “OS kernel”?

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

#6: 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)?

#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 is “preemptive scheduling”? How is it different from “non preemptive scheduling”?

#11: If a process scheduling algorithm is “non preemptive”, what problems are expected?

#12: What is “starvation” for processes?

#13: Which of the following short-term process scheduling will be the best in response time (average response time)? Justify your choice.

(a) FCFS (a. k. a. “FIFO”)
(b) Round robin
(c) SJF (a. k. a. “SPN”)
(d) SRTF
#14: Which of the following short-term process scheduling will be the best in throughput? Justify your choice.

(a) FCFS (a. k. a. “FIFO”)
(b) Round robin
(c) SJF (a. k. a. “SPN”)
(d) SRTF

#15: Which of the following short-term process scheduling will be the worst in avoiding “starvation” (i.e., “fairness”)? Justify your choice.

(a) FCFS (a. k. a. “FIFO”)
(b) Round robin
(c) SJF (a. k. a. “SPN”)
(d) SRTF

#16: Which of the following short-term process scheduling must be a pre-emptive scheduling? Justify your choice.

(a) FCFS (a. k. a. “FIFO”)
(b) Round robin
(c) SJF (a. k. a. “SPN”)
(d) SRTF

#17: What is “race condition”?

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

#19: What is “critical section”?

#20: What is “mutual exclusion”?

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

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

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

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

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

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

#27: What “signal” system call to a semaphore exactly performs?
#28: Why are the two system calls for semaphores (“wait” and “signal”) must be atomic operations (explain the reason)?

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

#30: Who manages semaphores?

#31: 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)?