

1. T/F Questions (2 points if correct, -2 point if not)

- (1) DMA is a mechanism for allowing an I/O device to transfer data to and from memory without involving the CPU in the transfer.
- (2) Memory mapped I/O determines how the pages of an I/O-bound process are mapped to page frames.
- (3) In round robin scheduling, it is advantageous to give each I/O bound process a longer quantum than each CPU-bound process (since this has the effect of giving the I/O bound process a higher priority).
- (4) It is not possible to implement user-kernel separation without hardware support for dual mode operation.
- (5) Shortest Remaining Time First is the best preemptive scheduling algorithm that can be implemented in an Operating System.
- (6) Paging leads to external fragmentation.
- (7) Threads within the same process share the same heap and stack.
- (8) When designing a multithreaded application, you must use synchronization primitives to make sure that the threads do not overwrite each other's registers.
- (9) When a process issues a system call, the OS code starts by executing an instruction to change the processor mode (from user to kernel).
- (10) Software interrupts are synchronous with the current process.
- (11) A context switch takes place at every system call.
- (12) Switching among threads in the same process is more efficient than switching among processes.
- (13) Using mutual exclusion ensures that a system avoids deadlock.

2. What are the two roles of an Operating System. For each role, briefly (one to two sentences) explain the role. (4 points)

3. As the price of memory is getting cheaper and cheaper, Mr. Mony Mana finally decided to buy a large amount of memory in order to avoid thrashing on his computer. Does this help to avoid thrashing? And how much memory should he buy to avoid it completely? (5 points)

6. Mr. Krazy Coder argues that applications can manage page tables better than OS, since applications know their own characteristics better. He then suggests each application should contain its own page tables inside the application itself, instead of inside the memory region of operating system. Does he make sense? (5 points)

7. Suppose a thread is running in a critical section of code, meaning that it has acquired all the locks through proper arbitration (중재). Can it get context switched? Why or why not? (3 points)

8. We consider a system consisting of two processes, P0 and P1, each accessing two semaphores, S and Q, set to the value 1.

P0	P1
Wait(S)	Wait(Q)
Wait(Q)	Wait(S)
...	...
...	...
...	...
Signal(S)	Signal(Q)
Signal(Q)	Signal(S)

What kind of unwanted situation will happen? Explain your answer. (4 points)

9. We consider a system consisting of two processes, P0 and P1, each accessing two semaphores, S and Q, set to the value 1.

P0	P1
Wait(S)	Wait(S)
Wait(Q)	Wait(Q)
...	...
...	...
Signal(S)	Signal(Q)
Signal(S)	Signal(S)

What kind of unwanted **situation(s)** will happen? Explain your answer. (4 points)