TI2 Flipped Classroom, The Correct Answers

Processes

How can a process be started?
Yes: The user starts the process in a shell or by clicking some icon
No: An interrupt occurs (e.g. the user presses a button on a disk drive) (although interrupt handlers may be understood as short-lived threads)
Yes: A process can create a new process
Yes: A process can fork itself to get two processes, a parent and a child
Action item: Explaining processes and fork()

What is a PCB?
Yes, chemistry: Polychlorinated biphenyl, organic compounds
Yes, CS (and this lecture): Process control block, an operating system data structure
Yes, EE: Printed circuit board, a board used in electronics
Action item: Explain PCB

Which of the following statements are true?
Yes: Number of Threads ≥ Number of Processes
No: Number of Processes ≥ Number of CPU Cores (usually yes, but in general no)
No: Number of Processes ≤ Number of CPU Cores
No: Number of Running Threads ≥ Number of CPU Cores
Action item: Explain three main states of thread

When does a thread stop using the CPU?
Yes: When it dies
No: When another thread claims its CPU
Yes: When it is waiting for I/O or a lock
Yes: When it wants to sleep
Yes: The OS can also decide to stop a thread!
Action item: Explain preemptive vs. non-preemptive OS

Based on which criteria does the OS choose the next thread to run?
Yes: Has high priority
Yes: Has not been running often recently
No: Has enough free memory
Yes: Holds an important lock
No: Will terminate quickly
Action item: Explain priority inversion

Inter-Process Communication IPC

Are these IPC Methods?
Yes: Pipes
Yes: Sockets
Yes: Signals (but almost no data)
Yes: Shared Memory
Yes: Windows Messages
Yes: Locks (using shared memory, or files, or whatnot)
Action item: Even more, e.g. remote procedure calls

How can two processes have access to the same shared memory?
Yes: Use shmget with the same key
Yes: A child process can access the shared memory of its parent
No: The use the same global variable names
Yes: Memory-map the same file
Action item: Sketch shmget

Concurrency (of Threads)

How can two threads concurrently (and correctly) increment a number?
No: Not possible at all
No: Just do it
No: Yes, using the "++" operator
Yes: Use Locks
Yes: Use Compare & Swap
Yes, but No: Use Peterson's algorithm (see next item)
Action item: Explain Peterson

Can Peterson's algorithm be used to implement a lock?
No: Yes
No: Yes: Only if the shared variables are marked "volatile"
No: No: Even though the variables were marked volatile, the compiler reordered the instructions
Yes: No: Even though the variables were marked volatile, the CPU reordered instructions
Action item: CPU reordering

Mutual Exclusion vs. Locks
Yes: Locks can be used to achieve mutual exclusion
Yes: Mutual exclusion is a way to implement a lock
Yes: Hardware primitives (e.g. test-and-set, swap) are usually used to implement locks
Yes: In Java, every object contains a lock
Action item: HW -> Locks -> Mutex etc.

What is the problem of the code in Figure 6.7?
No: There is a bug, it does not work
Yes: It works but it is unfair, as some thread may never get the lock
Yes: It works but it is inefficient

What is the advantage of a Semaphore over a Lock?
No: More powerful (can do what locks cannot)
Yes: A thread waiting for a semaphore can immediately be woken
Yes: Semaphores can directly provide access to more than just a single resource

Dining Philosophers ...
No: ... applies to Pizza eating
Yes: ... can be used to explain deadlocks
Yes: ... can be used to demonstrate parallelism