1. What information must the OS save when performing a context switch?
   1. Registers (stack pointer)
   2. Shared memory, caches (state of memory), PC, kernel pointers
   3. Context switch is expensive because cache is lost
2. Why does Shortest Time to Completion (similar to SJF) require preemption?
   1. Every time a process enters, need to check its time to completion and compare to the current process--> Switch to the shorter one.
3. What resources are replaced when exec is called?
   1. By the other program’s code.
   2. Machine’s state: the parent process state
   3. Stack and heap
4. True/False: race-conditions cannot arise on a single-core CPU because instructions can only be executed sequentially
   1. False. Multiple threads on the single core.
   2. True for single core and single thread.
5. Assuming the same single-core architecture, propose one change at the kernel level such that inter-thread race conditions will never occur.
   1. Switch to non-preemptive.
   2. Cooperative scheduling: no context switches at critical sections. Transfer control after done with critical section.
6. What is atomicity?
   1. All or nothing(data), before or after (multiple operations)