New York University

CSCI-UA.0202-003: Operating Systems (Undergrad): Fall 2025

Quiz 3

- Write your full name on both:
  - the bubble sheet in the "Name" field
  - the quiz booklet
- Write your NYU NetID on the quiz booklet and the bubble sheet in the "ID" field
- Use a #2 pencil to fill in your answers on the bubble sheet (preferred, but you can also use a pen)
- This quiz contains 6 questions only. Each question has choices from A to D
- Fill the bubbles completely by darkening the entire circle, as shown in the example
- Only mark answers for questions 1-6. Do not mark any bubbles beyond question 6
- Choose only one answer per question
- Submit your bubble sheet **together with your quiz booklet**

Name:

NetId:

1. According to what we discussed in lectures, why must a thread that calls `cond_wait()` use a `while` loop to re-check the condition upon waking?
   (A) To ensure the lock is held when calling `cond_wait()`, which is a requirement.
   (B) To handle spurious wakeups, where a thread might be awakened even if the condition it's waiting for is not yet true.
   (C) To prevent other threads from signaling the same condition variable simultaneously.
   (D) To guarantee that the thread with the highest priority is always awakened first.

2. In the context of a Monitor, what is the relationship between its mutex and its condition variables?
   (A) The mutex is only used to protect the condition variables, not the shared data.
   (B) A monitor can have multiple mutexes but only one condition variable.
   (C) The mutex that protects the monitor's methods is the same mutex that must be held when operating on its condition variables.
   (D) Condition variables operate independently and do not require the monitor's mutex to be held.

3. In the mutex implementation we introduced in the lecture, a thread attempting to acquire a held lock performs these steps (among others): adds itself to the `waiters` queue, calls `sched_mark_blocked()`, releases the internal spinlock, and finally calls `sched_swtch()`. Why is it critical to release the internal spinlock *before* calling `sched_swtch()` to go to sleep?
   (A) To allow the `sched_swtch()` function to reuse the spinlock for its own internal synchronization.
   (B) To ensure the thread still holds the spinlock when it wakes up, so it can resume safely.
   (C) To prevent the scheduler from immediately waking the thread back up again, which is known as a spurious wakeup.
   (D) To avoid no other thread could ever acquire it to modify the mutex's state (e.g., to release the mutex).

4. The entire reader-writer lock mechanism uses several components working together: integer counters (`AR` , `AW` , `WR` , `WW`), two condition variables (`okToRead` and `okToWrite`), and a single `Mutex` . The condition variables are responsible for putting threads to sleep and waking them up. Given that, what is the essential role of the single `Mutex` in this mechanism?
   (A) It serves as the primary lock on the database itself; no thread can read or write without holding this `mutex` .
   (B) It is a "master switch" that determines whether the lock should currently favor readers or writers.
   (C) It protects the integer counters (`AR` , `AW` , etc.) from race conditions, ensuring that when one thread is changing their values, no other thread can read or write them at the same time.
   (D) It is only used to prevent two threads from calling `wait()` on the same condition variable at the exact same time.

5. According to "An Investigation of the Therac-25 Accidents", what was the name of the company that manufactured the Therac-25?
   (A) General Electric (GE)
   (B) Atomic Energy of Canada Limited (AECL)
   (C) Siemens AG
   (D) Varian Medical Systems

6. "An Investigation of the Therac-25 Accidents" emphasizes that the Therac-25 incidents were "system accidents." Which of the following best describes what this means?
   (A) The accidents were caused exclusively by a single, catastrophic software bug that was impossible to detect.
   (B) The disasters resulted from a complex interplay of factors, including flawed software design, the removal of hardware safeguards, and poor organizational procedures.
   (C) The operators were the primary cause of the accidents due to their repeated failure to follow the correct procedures.
   (D) The machine's hardware was fundamentally unreliable and would have failed regardless of the software installed.