**Name: Martin L. Metke**

**Problem Number:** 1.A

Execution State:

Under normal circumstances, the “empty” semaphore keeps track of remaining unfilled slots in the buffer. Calls to empty.acquire() assure that an insert() call cannot proceed until at least one slot is open. Likewise, every remove() call uses empty.release() to increment the number of empty slots and allow new insertions to proceed.

Problem encountered:

Assuming that empty.release() from line 82 were deleted, it would eventually become impossible to insert new entries even if every existing entry were removed via the remove()

method. Continued calls to remove() would succeed, although they would return “null” rather than actual items as the out counter wraps around. However, all new insert() callers would starve.

|  |  |
| --- | --- |
| Process #0 – the “remove()” thread | |
| Line # | 82 ( was empty.release(); ) |
| Relevant Variables | |
| empty | This semaphore goes to 0 count when the buffer is full, and increments when an entry is removed. |

|  |  |
| --- | --- |
| Process #1 – the “insert()”thread | |
| Line # | 49 (empty.acquire(); ) |
| Relevant Variables | |
| empty | With calls to “remove()” no longer releasing the empty Semaphore counts, all subsequent calls to insert() will block indefinitely. |