

Operating Systems - Monsoon 2021

Sambuddho Chakravarty, Arani Bhattacharya

December 17, 2021

Assignment 4: Kernel Synchronization Primitives

Due: Jan. 3, 2022 @2359hrs. Late submission penalties apply

Kernel Producer-Consumer Problem (Total points: 40).

This exercise requires you to create a kernel driven producer and consumer. There should be two processes P and C. P reads random 8-bytes of the device `/dev/urandom` and passes them to the kernel via a system call, lets name it `writer()`. The kernel should maintain a queue of fixed length (allocated using `kmalloc()` function). The kernel implementation of the `writer()` should accept the 8-byte blocks from the user and write them the queue (and NOT a shared memory).

The process C should call a `reader()` system call, whose kernel implementation must involve calling the function to dequeue the enqueued bytes, from the read-write queue. You would therefore need functions to enqueue and dequeue bytes from the queue. These functions must use kernel synchronization primitives like kernel reader-writer semaphores or mutexes to ensure that there are no race conditions. As a suggestion, avoid using spin locks, until and unless you are really sure about what you are doing.

What to submit/[rubric](#).

1. Fully functional and modified kernel diff which can be patched to a stock kernel and used (preferably 5.10 upwards). The diff should correspond to the code that must include the system calls' implementations to read and write from the shared queue. Additionally you would require the programs P and C that write and read the bytes from the kernel queue respectively. The programs may print them to the terminal using `printf()` for the sake of convenience.
[35 points]. [Full compilation and correct functionality demonstrated using the test program \(35 points\)](#). [Patched kernel compiles successfully but meets only a few functionality requirements \(25 points\)](#). [Patched kernel doesn't compile or compiles but doesn't boot up. But the functionality and changes presented seem valid and correct, though cannot be validated \(12.5 points\)](#). [The patched kernel doesn't compile or compiles but doesn't boot but the functionality doesn't seem correct, let alone be validated \(0 points\)](#).
2. A readme describing the logic of the programs with relevant description of the data structures used, including description of the kernel data structures and source files modified (which ones were modified, how and why) **[5 points]**.