Joshua Andrews

April 19th 2018

ECE-331

Introduction

This report provides a brief overview on the purpose of locking in the linux kernel. Then a discussion on the design and testing of locking for the four-fsk driver is followed by a conclusion.

Locking: Purpose, Design, and Testing

Locking is used in the linux kernel for concurrency management. Without locking, race conditions can occur due to shared access of the device, causing unexpected results. A mutex is used to provide locking in the four-fsk driver write function. The mutex, which is based on semaphores, prevents other processes from initiating a write command to the four-fsk device if it is currently in use. If the non-blocking flag is set for opening the four-fsk device and the device is locked, an error is returned. If the flag is not set, the blocked process will go to sleep and wait until the lock is released. Careful attention must be paid to ensure that when a lock is set, it is subsequently unlocked to avoid a deadlock, which disables write access to the device. The mutex is interruptible, allowing the user to cancel a waiting process. To test for proper operation of locking in the four-fsk driver, the fork function was used to generate multiple processes. These multiple processes then simultaneously write to the device. With non-blocking specified, one process was able to write to the device and all others were canceled. With blocking, the processes wait and write sequentially to the device.

Conclusion

The design and testing of locking was discussed. Locking in the driver worked correctly and was verified by a userspace program and the usb serial output of the expansion board.