

CECS 326-01

Operating Systems

Connor McKenna 031658430

Assignment 4

Due Date: 11/19/2024

Submission Date: 11/20/2024

Program Description

1. Unity
 - a. The programs demonstrate how multiple processes can communicate and share data safely using shared memory and semaphores. The master program creates a shared memory segment and spawns multiple slave processes. Each slave writes its unique process number to the shared memory. A semaphore is used to ensure only one process writes to the shared memory at a time, preventing conflicts. After all slave processes finish, the master program cleans up the shared resources.
2. What each program does individually.
 - a. `master.c`

The master program creates and initializes a shared memory segment, including an index to track available slots. It also creates a named semaphore to ensure mutual exclusion when processes access shared memory. The master spawns a specified number of slave processes, waits for them to finish, and then displays the content of the shared memory. Finally, it removes the semaphore and shared memory segment before exiting.

b. slave.c

Each slave process receives its child number and the shared memory name from the master. It writes its number to the next available slot in the shared memory and increments the index, ensuring mutual exclusion by locking and unlocking the semaphore. Once finished, the slave process releases its access to the shared memory and terminates.

c. myShm.h

The myShm.h header file defines the structure used for shared memory. It contains an index to track the next available slot and a report array where each slave process writes its number. This file ensures consistent definitions across master and slave programs.