CECS 326-01 Operating Systems

William Luu

Assignment 3

Due Date: 10/29/2024

Submission Date: 10/28/2024

Program Description

The two programs master.c and slave.c as well as a header file myShm.h use previously learned system calls as well as the POSIX implementation of Linux's shared memory mechanism. The program starts with the master executing, creating a shared memory segment utilizing myShm.h's CLASS structure. This structure holds an index and a report which is an integer array. The master creates, executes, and waits for child processes to finish, where the child processes are from slave.c. The slave processes will then write their child number into the shared memory segment slots and terminate, where master then prints out the contents of the shared memory segment and then terminates.

master.c

The master.c program implements a process that creates a shared memory segment and spawns a specified number of child processes. It starts by checking command-line arguments to ensure the correct number of parameters are provided. Upon execution, the master outputs its status and creates a shared memory segment using the POSIX shmget() function, attaching it for use with shmat(). It initializes the index of the shared memory structure to zero before forking child processes, each of which executes the slave program with its respective child number and the shared memory segment name. After creating all child processes, the master waits for their termination, then outputs the contents of the shared memory segment, displaying the child numbers recorded by the slave processes. Finally, the master detaches from and removes the shared memory segment before exiting.

slave.c

The slave.c program serves as the child process in the shared memory setup initiated by the master.c program. Upon execution, it first verifies that the correct number of command-line arguments has been provided, then prints its child number and the name of the shared memory segment it received. The slave process opens the existing shared memory segment using shmget() and attaches to it with shmat(). It writes its child number into the report array at the current index of the shared memory structure, provided there is space (i.e., the index is less than 10). After updating the index, it detaches from the shared memory segment and outputs a message indicating that it has closed access before terminating.