

CECS 326-01 Operating Systems

William Luu

Assignment 4

Due Date: 12/05/2024

Submission Date: 12/05/2024

Program Description

This program is an enhanced version of the program from Project 4. Similarly, we are using the master program that creates a shared memory segment in which child processes created and executed by the master program will access. These child processes are called slave processes that will use an unnamed semaphore to take turns accessing the shared memory segment. A named semaphore is implemented and used to ensure that the slave processes' outputs will be in a distinct order, to prevent any race conditions.

master.cpp

Similarly to the previous chapter, the master program takes arguments from the command line, which are the number of child processes to be created and the name of the shared memory segment. The master process will then access the data structure that holds the shared memory segment, and initialize the report to be zeros. An unnamed and named semaphore will be initialized as well as the child processes. These child processes will be executed, while the master process will wait for all child processes to terminate. All semaphores and shared memory segments will be unlinked and removed once all child processes terminate.

slave.cpp

The slave process, with addition to the previous implementations of Project 4, now has a named semaphore to control its outputs. Each slave process will create an output file, where the outputs of the slave process will be written and stored. A name semaphore will be opened and ready to use. The slave process's critical zone of outputting the output file into the terminal is controlled by the named semaphore. After outputting the necessary outputs, the slave process will delete the created output files from the directory, detach from the shared memory and then terminate.