CSE344 - HW03 Report

Abdullah Çelik

April 2022

1 Overview

- First, all requirements are done.
- Error handling of command line arguments is done.
- Shared memory was used for the unnamed semaphore part. Semaphores and some variables are in shared memory.
- Shared memory was used for some variables in the named semaphore part. Semaphores are not placed in shared memory.
- 12 processes were created. These are main process, wholesaler process, 6 chefs and 4 pushers. A pusher was created for each ingredients.
- Main process create wholesaler process, wholesaler creates chefs and pushers processes.
- The main process reads from the given input file and fills it into the character array where in the shared memory. It then wakes up the wholesaler who in the block queue. The wholesaler reads ingredients from the shared memory and wakes the pushers who in the block queue for each ingredients. After the pushers do sem_post process according to the incoming materials, the chef who can cook starts the cooking. Then chef finishes cooking and wakes up the wholesaler who is waiting in the block queue and tells him that the cooking is done. If the main process cannot read any more wholesaler terminates the pushers with a signal. The wholesaler organizes the field in the shared memory and wakes the chefs from the block queue. By checking this field, the chefs realize that there are no more ingredients and return the desserts they have cooked. The wholesaler collects these values with waitpid system call and prints it to the screen.
- The reason why the chefs were forked by the wholesaler is to create a parent child relationship between them and send the total cooked desserts without using pipes, shared memory or anything else.

- In case of SIGINT, all children forward SIGINT signal to the main process. The main process handle it and sends SIGUSR1 to all child processes and terminates itself. The children terminate itself because of the incoming SIGUSR1. Before terminating, all resources are returned to the system, opened files are closed, the created shared memory is closed and unlinked.
- In the named semahore part, semaphore names are built from the semaphoreName given as an command line argument. Their name structures is "<semaphoreName> <integerValue>".

2 Compile and Run

- make → Compiles the whole program
  Type make in the file contains the makefile
- make clean → Cleans all objects files
- ./hw3unnamed -i inputFile → Runs unnamed semaphore part
- ./hw3named -i inputFile -n semaphoreName → Runs named semaphore part