# CSE344 Systems Programming Report

Tolga Reis, 161044035

Homework 4

## 1 Introduction

Each chef has to check two ingredients in common area so I used System V semaphores since supports semaphore sets and POSIX semaphore as mutex. To provide synchronization semaphore set includes four different semaphores and they represents synchronization for “milk”, “flour”, “walnut”, and “sugar”.

## 2 Main Thread(Wholesaler)

- Main function runs as a main thread and critical section part of the main represents “wholesaler”.
- In the main SKEY is used to create System V semaphore set. It is created during the program and deleted after all operations are done.
- semctl runs four times for four different semaphores and SEM-NUM macro constant represents semaphore number.
- THREAD-NUM is a macro constant and represents total number of chefs. pthread-create creates six different threads with chef thread function and gives their order with chef-num heap array. This heap is freed at the end of the main.
- srand() is called to create random numbers after that. POSIX mutex is initialized.
- common-bag heap array is initialized to be accessed by all threads. read-buffer is initialized to read ingredients from file.
- semaphore-buf is created to provide semaphore set with four semaphores. Their flags are initialized to 0 for operation.
- Then the loop starts. At each iteration three bytes are read, first and second characters represent ingredients; the last one is ‘’ character and ignored. If there is not enough characters or it reads EOF breaks the loop.

- Then selection variable is initialized according to select-chef function return value. select-chef decides which chef is selected according to lack of ingredients. This selection provides post for needed semaphores. Also read ingredients are written to common-bag to be read from threads.
- Semaphore operations are initialized for all semaphores and according to selection, sem-nums are assigned. After that wholesaler prints delivering message, posts the semaphore set. After posting prints the waiting message, waits mutex semaphore.
- After waiting the mutex, loop iteration returns to beginning.
- After all ingredients are delivered, posts all semaphores to notify chef threads, then waits for all threads to exit.
- At the end frees all memory allocations and unlinks the SKEY.

# 3 Chef Thread

First of all, thread argument is casted to integer to use as chef number. Then sembuf is created for synchronization semaphores. But this time synchronization is as wait. According to chef-num, sem-nums are assigned. After semop, end condition is checked. At the end of the loop, take messages are printed. And mutex is posted. After the loop pthread exits.