CSE 344 HOMEWORK 4 REPORT

Furkan Kalabalık - 161044001

Gebze Technical University
Computer Science and Engineering Department

24 May 2020 9:00 AM

In this homework we're asked to implement a wholesaler waits for güllaç and 6 chefs that waits for ingredients to prepare güllaç. To prepare güllaç chefs need 4 ingredients: milk(M), flour(F), sugar(S), walnuts(W). Every chef has two of these ingredients infinitely. Purpose of wholesaler comes here. To prepare güllaç chefs needs two more ingredient and wholesaler gives these ingredients to chefs. It serves two distinct ingredients at a time without knowing anything about chefs and wait for one them, to prepare güllaç. Then wholesaler takes desert to sell it. We need to implement this problem with POSIX threads and ingredients will be taken from a input file.

We need to create only one function that implement all chefs and a main thread that acts as wholesaler. Also all chefs must use same semaphores. We need to make sure that wholesaler must serve two distinct item at a time to a shared data structure at heap so all threads can access it. We need make sure that all chef threads joins the main wholesaler thread and in any case of exit or finish, process must exit gracefully. Also whole must not be aware of number of chefs.

First, I create command line parser and take the input file name properly. After that to make sure file format is appropriate, I checked file by following these rules:

- Pair of ingredients must be served in one line
- Every pair must be distinct so one chef can prepare güllaç.
- File must contain at least 10 service of ingredients.
- If there is a unknown ingredient or if there is 3 ingredient at one line, we print a error message and exit.
- Example file format be like:
  - MS
  - MW
  - FM
  - and so on.

If file is appropriate then we can proceed the next step. For the purpose of same semaphore usage, I used a mixed formula. When wholesaler served ingredient, to wake up only the related chef I created a systemV semaphore set with 4 item that corresponds every ingredient. This set is initialized with all 0's at main thread before serving. Also to wait the wholesaler when desert preparing, I used a POSIX semaphore as binary semaphore called gullacLock that is initialized to 0. Also, I registered a cleanUp function in case of exit that closes input file and destroys semaphores.

From there I created threads with pthread_create function. When I create threads I send function name that thread will execute called chefThread and a parameter integer as a void pointer. This integer indicates ID of chef. In function to print appropriate messages and wake up related chef, I used that id in a if else statement. I take return status of create function make sure that all threads are created safely. ID's sended as for example for chef1 I send 1 as ID and so on.

Now we can start serving ingredients. First, I assigned semaphore set items to every ingredients. Milk is the 0 semaphore of set, sugar is 1, flour is 2 and walnuts is 3. Then wholesaler reads file and put ingredients to heap structure. I used a array with 2 sized. I read 3 byte to pass newline char. After readed chars putted on heap array, I checked which ingredient pair wholesaler is readed. For example if wholesaler reads MF then following operations will occur:

- Wholesaler increase milk and flour on semaphore set 2 times and other two elements sugar and walnuts one time. By doing that only the chef that decrease these numbers can wakeup and prepare güllaç. All chefs waits for that semaphore and only the missing one that means chef that doesn't has two added ingredients will be wake up.
- After wholesaler served readed ingredients to data structure, it operates given values on semaphore set, and try to decrease gullacLock POSIX semaphore by applying sem_wait function. This means wholesaler will wait until a chef take ingredients, prepare güllaç and post gullacLock. After gullacLock is posted, wholesaler begins reading next pair of ingredients.

If there is no ingredients remaining, wholesaler stops reading and starts to kill chefs. It does it by poisoning chefs. First wholesaler puts P on shared structure(array). This represents poison. After that it op-

erates post operation on semaphore set by increase every set element adding 2. By doing that it can wake up any chef thread and that thread can die by taking poison. After a chef thread takes poison it joins with wholesaler thread by returning its ID.

In case of chefs, they have also a systemV operation set inside it, that associate with ingredients like wholesaler. Every chef tries to decrease semaphore set. They try to decrease missing ones 2 times and other ones 1 time. If only all conditions satisfied, decreasing will occur. Otherwise chef remains as waiting. It is just like a handshaking between wholesaler and chefs. Wholesaler doesn't know how many chef, just sends ingredient with indicates sended elements increasing 2 times. After they acquire accessing ingredients, chef looks for ingredients is right. In case of poisoning, thread can also acquire as normal ingredient and when chef looked ingredient is P then it gives back to resources that acquired from semaphore set and break its loop, terminates by returning its ID. If there is no P in delivered items, then it starts printing taken items and prepares desert by simulating process of preparing with a sleep that sleeps between 1-5 second. After desert is ready, related chef posts gullacLock and begins wait, by that time wholesaler takes desert and begins sending next pair of ingredients.

After all ingredients served, and all threads terminated, chefs joins main wholesaler thread and prints its ID's.At the end exit will starts clean up process and removes resources.