

**GTU Department of Computer Engineering**  
**CSE 344 - Spring 2022**  
**Homework #4 Report**

**Hasan Mutlu**  
**1801042673**

## HOW I SOLVED THIS PROBLEM

I have solved this problem as described in the PDF. The usages of the semaphores were already defined.

In order to count the number of delivered materials in supplier thread, I used local variables. For each character read, local variables are incremented and compared with semaphore's current values.

In order to synchronize the terminations of main process and the supplier thread, I used the same semaphores that are used for characters.

## DESIGN EXPLANATIONS

### Main Process:

- ➔ Main process initializes the semaphores with the value 0. Then it first creates the consumer threads, then the supplier thread.
- ➔ After that, it waits for consumer threads to join. Once all consumers join, it makes a wait operation on the semaphores.
- ➔ After all consumer threads join, semaphores' values are supposed to be 0. In order for main process to continue, the supplier thread must post the semaphores.
- ➔ After it continues, it frees resources, closes files, removes semaphores and terminates.

### Consumer Thread:

- ➔ In a loop for N times, consumer thread waits for semaphores.
- ➔ When it continues, it prints consumed message and continues to loop.

### Supplier Thread:

- ➔ Supplier thread has 4 local variables for counting the delivered characters. Two of these variables counts how many characters are read from file (read1, read2), and the other two counts total number of delivered characters (totdel1, totdel2).
- ➔ In a loop for  $2*N*C$  times:
- ➔ It first checks if any characters are delivered. If semaphores' values are lower than the number of read characters, it means that characters are delivered, as much amount as their difference. In loops where number of read characters (read1 and

read2) are decremented until they are equal to semaphores' current values, it increments total number of delivered characters and prints the delivered message.

- ➔ Then, it reads a character from the file, posts according semaphore and increments its read number value.
- ➔ After loop ends, it checks if any characters are delivered one last time.
- ➔ Finally, when it is done, it posts the semaphores' so that the main process can continue.

### **ACHIEVED AND FAILED REQUIREMENTS**

As far as I tested, all requirements are achieved.