

Gebze Technical University  
CSE344 Systems Programming Course

-

## MIDTERM REPORT

Muhammed OZKAN 151044084

May 2, 2021

### Abstract

This is basically a (compound) producer/consumer paradigm. We will emulate a covid19 vaccination flow. Nurses bring vaccine products to a clinic/buffer, vaccinators vaccinate the citizens by taking two of them (both the first and second shots) and the citizens leave the clinic. Each nurse, vaccinator and citizen will be represented by a distinct process. Each citizen needs to get the two shots  $t$  times.

## 1 Overview

The homework I submitted is a complete assignment. It supports all the requirements and parameters requested in the homework document + BONUS PART. It opens the file(vaccines) with the given path. Process reads the given parameters. It binds the shared memory space to itself according to the name specified in the code. Creates semaphores that will be used for synchronization in program flow.(for 1 Shared Memory, for 3 Producer/Consumer) It uses total 4 semaphore. After these processes, it creates the PIPEs required for communication. Child processes as many as specified in the parameter are created with the help of fork(). Then the main process begins to wait for the child processes to be done process. After all operations are completed successfully, it prints the closing messages on the screen and returns all the resources it uses and creates to the system.

## 2 How did I solve this problem.

In order to solve the problem, first of all, the parameters sent to our program had to be properly separated and analyzed. I used `getopt()` function in unix standard c library for this. By doing this, I was able to parse without any problems regardless of the order of the parameters sent. I checked whether the values I received with the `getopt` function meet the criteria of our program. I have reported the values that are not suitable or incorrect to the user with the help of `stderr`. In this way, the first phase of our program has been completed. All processes use a common global variable file descriptor. The program checks the suitability of the file read at startup.

Secondly, I created the semaphore and shared memory to be used in the program to protect the synchronization and shared memory part. I am keeping a struct structure in shared memory. I made the initial value assignments of the variables that should be kept in this common area. I created PIPE them because It communicate between Vaccinator and Citizin via PIPE. After this part, I created Citizen Vaccinator and finally Nurse child processes. After this section, the processes proceed over the functions set according to the type of each process.

## 3 My design decisions.

There are 3 different processes in the program: Citizen Nurse Vaccinator. The common structures they use are: Shared Memory, Empty, First, Second Semaphore. A different form of the classic Producer / Consumer approach is used in our design. The empty semaphore is initially initialized with the size of the buffer. The full semaphore has been used as 2 separate semaphore as the first semaphore and the second semaphore. Nurse reads each vaccine with wait from the empty semaphore , and posts the first or second semaphore depending on the type of vaccine.

On the other hand, the vaccinator waits for the First semaphore and the Second semaphore, respectively, and when it receives both, it selects a person from shared memory to inoculate that person. The selected Citizen is notified via PIPE. This process is done from the oldest to the youngest. It opens a new space by posting the empty semaphore. While all these processes are happening, the data in the common memory is controlled by a separate semaphore.

After the nurses carry the required amount of vaccine to the buffer, they give the information message and end. Vaccinators end after vaccinating all citizens with the vaccines in the buffer. Citizens end when they get the amount of vaccine they should be.

## 4 What requirements did I meet and which did I fail?

All requirements are fully met + BONUS PART. All processes created at the end of the program are terminated and resources are returned to the system. In case of CTRL-C the program stop execution, return all resources to the system and exit with an information message. All error messages print to stderr. All system calls are checking for failure and notify to user accordingly.

However, in the PIPE system I use, since each PIPE uses 2 file descriptors and the file descriptor that a process can open is limited to 1024 as standard, it gives a Too Many Open Files error during the pipe creation phase after 500 Citizens. We can increase this problem with the command "ulimit -n 4096". In this way, we can create PIPEs up to 2000 Citizens and test our program.

## 5 Tests

```
./program -n 2 -v 3 -c 3 -b 7 -t 2 -i inputfilepath
Welcome to the GTU344 clinic. Number of citizen to vaccinate c=3 with t=2
doses.
Nurse 1 (pid=53) has brought vaccine 1: the clinic has 1 vaccine1 and 0 vaccine2.
Nurse 2 (pid=54) has brought vaccine 1: the clinic has 2 vaccine1 and 0 vaccine2.
Nurse 1 (pid=53) has brought vaccine 1: the clinic has 3 vaccine1 and 0 vaccine2.
Nurse 2 (pid=54) has brought vaccine 2: the clinic has 3 vaccine1 and 1 vaccine2.
Nurse 1 (pid=53) has brought vaccine 2: the clinic has 3 vaccine1 and 2 vaccine2.
Nurse 2 (pid=54) has brought vaccine 2: the clinic has 3 vaccine1 and 3 vaccine2.
Vaccinator 1 (pid=50) is inviting citizen pid=47 to the clinic
Citizen 1 (pid=47) is vaccinated for the 1 time: the clinic has 2 vaccine1 and 2 vaccine2
Vaccinator 2 (pid=51) is inviting citizen pid=48 to the clinic
Citizen 2 (pid=48) is vaccinated for the 1 time: the clinic has 1 vaccine1 and 1 vaccine2
Nurse 1 (pid=53) has brought vaccine 1: the clinic has 2 vaccine1 and 1 vaccine2.
Nurse 2 (pid=54) has brought vaccine 1: the clinic has 3 vaccine1 and 1 vaccine2.
Vaccinator 3 (pid=52) is inviting citizen pid=49 to the clinic
Citizen 3 (pid=49) is vaccinated for the 1 time: the clinic has 2 vaccine1 and 0 vaccine2
```

Nurse 2 (pid=54) has brought vaccine 1: the clinic has 3 vaccine1 and 0 vaccine2.  
 Nurse 1 (pid=53) has brought vaccine 2: the clinic has 3 vaccine1 and 1 vaccine2.  
 Nurse 2 (pid=54) has brought vaccine 2: the clinic has 3 vaccine1 and 2 vaccine2.  
 Nurse 1 (pid=53) has brought vaccine 2: the clinic has 3 vaccine1 and 3 vaccine2.  
 Vaccinator 1 (pid=50) is inviting citizen pid=47 to the clinic  
 Citizen 1 (pid=47) is vaccinated for the 2 time: the clinic has 2 vaccine1 and 2 vaccine2. The citizen is leaving. Remaining citizens to vaccinate: 2  
 Vaccinator 2 (pid=51) is inviting citizen pid=48 to the clinic  
 Citizen 2 (pid=48) is vaccinated for the 2 time: the clinic has 1 vaccine1 and 1 vaccine2. The citizen is leaving. Remaining citizens to vaccinate: 1  
 Nurses have carried all vaccines to the buffer, terminating.  
 Vaccinator 3 (pid=52) is inviting citizen pid=49 to the clinic  
 Citizen 3 (pid=49) is vaccinated for the 2 time: the clinic has 0 vaccine1 and 0 vaccine2. The citizen is leaving. Remaining citizens to vaccinate: 0  
 All citizens have been vaccinated.  
 Vaccinator 1 (pid=50) vaccinated 2 doses. Vaccinator 2 (pid=51) vaccinated 2 doses. Vaccinator 3 (pid=52) vaccinated 2 doses. The clinic is now closed. Stay healthy.

```
./program -n 2 -v 2 -c 3 -b 4 -t 1 -i inputfilepath
```

Welcome to the GTU344 clinic. Number of citizen to vaccinate c=3 with t=1 doses.

Nurse 1 (pid=25522) has brought vaccine 1: the clinic has 1 vaccine1 and 0 vaccine2.  
 Nurse 2 (pid=25523) has brought vaccine 1: the clinic has 2 vaccine1 and 0 vaccine2.  
 Nurse 1 (pid=25522) has brought vaccine 1: the clinic has 3 vaccine1 and 0 vaccine2.  
 Nurse 2 (pid=25523) has brought vaccine 2: the clinic has 3 vaccine1 and 1 vaccine2.  
 Vaccinator 1 (pid=25520) is inviting citizen pid=25517 to the clinic  
 Citizen 1 (pid=25517) is vaccinated for the 1 time: the clinic has 2 vaccine1 and 0 vaccine2. The citizen is leaving. Remaining citizens to vaccinate: 2  
 Nurse 1 (pid=25522) has brought vaccine 2: the clinic has 2 vaccine1 and 1 vaccine2.  
 Nurse 2 (pid=25523) has brought vaccine 2: the clinic has 2 vaccine1 and 2 vaccine2.  
 Vaccinator 2 (pid=25521) is inviting citizen pid=25518 to the clinic  
 Citizen 2 (pid=25518) is vaccinated for the 1 time: the clinic has 1 vaccine1 and 1 vaccine2. The citizen is leaving. Remaining citizens to vaccinate: 1  
 Vaccinator 1 (pid=25520) is inviting citizen pid=25519 to the clinic

Citizen 3 (pid=25519) is vaccinated for the 1 time: the clinic has 0 vaccine1 and 0 vaccine2. The citizen is leaving. Remaining citizens to vaccinate: 0  
Nurses have carried all vaccines to the buffer, terminating.  
All citizens have been vaccinated.  
Vaccinator 1 (pid=25520) vaccinated 2 doses. Vaccinator 2 (pid=25521) vaccinated 1 doses. The clinic is now closed. Stay healthy.