Assignment 3 – Threads, Processes and Semaphores

Assignment Type: Advanced Assignment

Deadline: Thursday, 2nd of April 11.55 PM, 2020.

Task 1: COVID-19 Management System

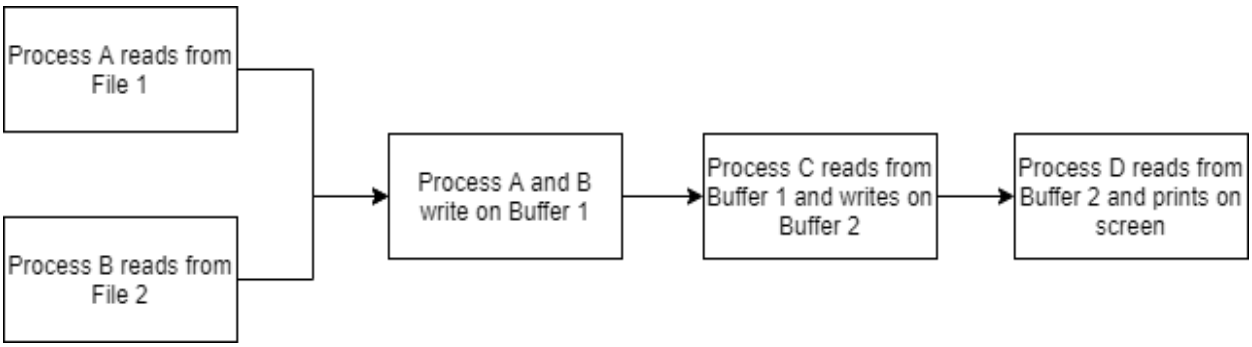
Help the world in crisis by developing a Coronavirus (COVID-19) management system.

You are required to manage the number of coronavirus patients via threads and semaphores. Suppose N number of potential Corona patients enter the hospital. You will have to write a program that initializes N threads simultaneously where each thread increments a shared variable **potentialCPatients**. After tests are being carried out for each potential patient, a patient may either be infected or not infected by coronoavirus in a random fashion. If the patient is infected, a semaphore called **coronaPatient** will be *signaled/incremented*. Whereas if the patient is NOT infected, a semaphore called **fluPatient** will be *signaled/incremented*. In either case, the shared variable called **potentialCpatient** variable will be decremented. You are not concerned with the recovery of the flu patients but only with the coronavirus patients. Eventually, each corona patient has to get better. For this purpose, each corona patient will go through tests and *wait* will be called for the **coronaPatient** only after two consecutive tests return false.

NOTE: Create necessary functions to enhance code readability. Also, N must be taken as user input.

Task 2: Process Synchronization and Order

Four processes are involved in printing files characters (pictured below). Using semaphore(s), order the execution of these processes such that first 10 characters are printed from each file 1 and 2. All four processes are spawned by the parent process at the same time. However, the execution of process D waits for the execution of process C. Whereas the execution of process C waits for the execution of process A and B. Also, remember that process A and B can both write on the same Buffer-1. Therefore, data must NOT be overwritten on Buffer-1 and synchronization must be ensured on Buffer-1 between process A and B.



Programming Language

You are required to complete this assignment using the C language. Make sure to compile your code using a C and not a C++ compiler (use .c as extension and not .cc or .cpp). This means that you cannot use C++ features such as classes, virtual methods and cout and cin for I/O. Some notes:

1. Instead of using cin and cout for I/O, use the printf and scanf functions.
2. To dynamically allocate memory, use the malloc and free functions instead of new and delete.
3. A man page exists about every function in the standard C library. For example, to learn more about scanf use man scanf. The manual pages about library functions are always in section 3: man printf will give you information about the shell command, but man 3 printf about the C library function. Similarly, system calls are in section 2. 2
4. Do not include iostream or set a namespace. Instead, include <stdio.h>, <stdlib.h>, <string.h> and <unistd.h>.
5. Ask the teaching assistant for help if you have problems!

GitHub Submissions:

This assignment is to be submitted on Github as well as SLATE.

Make sure you add the Teaching Assistant as collaborator on each assignment on github.

Username: MuhammadAbdullahAziz98

For any queries, consult the TA: Mr. M. Abdullah Aziz via email: l164085@lhr.nu.edu.pk