CSC 332 - Operating Systems

Lab – Spring 2021

Lab 5 - Process Synchronization

Dad-Son Problem

Max Points: 30 **Due**: April 22, 2021

The given bank.c program has 3 processes namely, the *dad* process and two *son* processes. The critical section (CS) in the given problem is a son withdrawing money from the bank and the dad depositing money in the bank, at randomly selected time intervals. The program in the given form has synchronization errors, i.e., these 3 processes get into a race condition when accessing the shared bank_balance variable. The program itself compiles correctly (i.e., there are no syntax errors).

In **Step 1**, you need to run the program and analyze the execution traces, whereupon you have to identify the synchronization errors. After identifying the synchronization errors, you need to insert the "P(sem)" and "V(sem)" operations at the right places in the code that fix the synchronization errors.

In **Step 2**, you work on a measurement component. A mutex algorithm is associated with a "bounded wait" property, i.e., how long a process P is forced to wait for entry into a CS after P has expressed interest in the CS. Let’s call the wait time T(P), where T(P) is the number of times processes other than P enter the CS after P expresses interest in the CS but before P actually enters the CS. You need to compute T(P) for all three processes in the problem and display it at the end of "N" different attempts. Measure T(P) for at least 5 different N’s and include the values in your report.

Instructions

- There are two different interfaces to semaphores in the POSIX standard: POSIX 1003.1b semaphores and System V semaphores. In this lab, we will use System V semaphores. The lab drive folder has the file “sem.h” that contains the usual create, destroy, P (or wait), and V (or signal) operations for System V semaphores which would be used to fix synchronization errors.
- You need to have sem.h header file in your present working directory to run this program and invoke semaphore operations.
- The bank.c file is documented. Once you are done with your solution, insert comments at the places where you made changes to fix the synchronization errors and add the wait time measurement.
- This synchronization problem should be solved with as few semaphore variables as possible.

Submission Instructions

Submit the bank.c file with semaphore operations inserted at appropriate places and measurement components. Write a brief report (at most 1 page) on how your solution prevents race conditions between the dad and son processes and measure the wait time of each process. Include both the files into a single folder and upload to the drive***