CS69003 Computing Systems Lab – I
Autumn 2007

Use of semaphores

Due: September 13, 2007

Part 1

Synchronize the two programs in Assignment 5 using a semaphore (instead of the *done* variable). Name the files A_1.c and B_1.c.

Part 2

Consider a file containing student records (maximum 100) in the same format as specified in Assignment 2 earlier. A process X, when started, first loads all the records in shared memory. It then goes to sleep, waking up periodically (say every 5 seconds) to check if any of the data have been modified (how will it know that?). If yes, it writes the entire content of the shared memory back to the file. The process X runs forever.

Any number of other processes can query and update the data by reading/writing the shared memory. Specifically, a process Y is allowed to do the following operations on the data:

- Search for a student record by roll no. (should show all data for that student)
- Update the CGPA of a particular student (the student is identified by the roll no., which is unique for each student)

The process Y will keep on doing the above two in a loop until the exit option is chosen (same as Assignment 2). Note that any number of copies of Y can be started at one time (think of this as multiple people searching/updating student records at the same time). Also, the following criterions must be ensured:

- If any copy of Y is started before X is started, it should wait for X to be started first.
- Only X should create and initialize all necessary shared memory.

Write two C programs, X.c for the process X and Y.c for the process Y. Your programs must be well-commented to make it easier for anyone to understand what semaphores you are using and how they are used.

Submission

Submit the file A_1.c, B_1.c, X.c, and Y.c. Tar all the programs into a single tar file Assgn6.tar and mail it to course email together with the subject line "Assignment 6 for 07CS60**".