CS69003: Computing Systems Lab I
Autumn 2008

Assignment 5

Use of shared memory and semaphore

Due: September 12, 2008 (Friday)

In this exercise, you are supposed to implement a dynamic search structure to be shared by multiple processes running on the same machine. In conformance with your earlier assignments, use a hash table of names with open addressing and linear probing (see Assignment 1).

Part 1: Use of read-only shared memory **(40)**

Write a program `inittbl.c` that creates a shared memory segment of an appropriate size for storing a hash table of names, reads the text file `names.txt`, and populates the hash table by the names stored in the file. Use open addressing and linear probing. The size of the hash table may be restricted so as to store one hundred names only.

Write another program `accesstbl.c` in order to search for names in the hash table stored in the shared memory. You may assume that the initialization process runs and initializes the shared memory earlier than the accessing process starts running.

Part 2: Synchronization of read/write shared memory by semaphores **(50)**

Modify the program `inittbl.c` so that it creates, in addition to the shared memory, a semaphore for synchronized access of the shared memory. The shared memory should be kept locked during the time period when the initialization process reads the input data file `names.txt`. After the hash table is populated by the entire input database, the semaphore should be freed.

Modify the program `accesstbl.c` so that it can insert new names to the shared hash table and delete existing names from the hash table. During each insert/delete operation, the shared memory should be locked by the accessing process.

In this part too, you may assume that the shared memory and the semaphore are initialized appropriately before any accessing process starts running.

Part 3: Synchronization of the creation of shared memory and semaphore **(10)**

Modify the two C files `inittbl.c` and `accesstbl.c` so as to handle the situation where an accessing process runs before the initialization process gets a chance to create the initial hash table in the shared memory. In this situation, the accessing process should quit after reporting an error message like “Data not available”.

You are required to submit two C files `inittbl.c` and `accesstbl.c` solving all the three parts. Mention your name and roll number (and also the assignment number) in the subject of the e-mail. As the initial database, you may use the file `names.txt` supplied in connection with Assignment 1.