

CS69003 Computing Systems Lab – I

Autumn 2007

Use of shared memory

Due: September 5, 2007

Part 1

Write 2 C programs *A.c* and *B.c*. The program *A.c* takes a command line argument *inppfile*, and reads in a set of integers (maximum 100) from the file named *inppfile* and writes it to a shared array. The program *B.c* takes a command line argument *outfile*, reads the set of integers from the shared array (how will it know how many integers are there?), sorts it (use quicksort), and prints the sorted output in a file named *outfile*.

Make sure that *B.c* deletes all shared memory created before exiting.

There is an obvious synchronization problem here, *B.c* should not start until *A.c* has finished writing the integers in the array. For the first part, ignore it, and start the program for *B.c* a few seconds after *A.c* starts.

Part 2

In this part, we will try to synchronize *A.c* and *B.c* by a simple method. Create a shared integer variable called *done* and initialize it to 0. *done* = 0 indicates that *A.c* has not finished writing the integers into the array. The program in *A.c* sets *done* to 1 after it finishes. The program in *B.c* periodically checks *done* and loops until it is 1. Modify *A.c* and *B.c* to implement this.

Is this scheme guranteed to work always?

Submission

Name your programs *A_1.c* and *B_1.c* for the first part of the assignment, and *A_2.c* and *B_2.c* for the second part. Tar all the programs into a single tar file *Assgn5.tar* and mail it to course email together with the subject line “Assignment 5 for 07CS60**”.