Assignment 4

EEL4930/5934 Advanced System Programming

Due: Monday, March 6th by midnight

In this assignment, you are going to simulate electronic fund transfer (EFT) between bank accounts the same way you did for Assignment3. However, in this assignment instead of having multiple threads you will have multiple child processes as the workers that perform the transfers concurrently.

You will be using the mmap() system call to let the worker (child) processes share memory. You are encouraged to use Pthreads Library's synchronization constructs such as mutexes and condition variables or the semaphore library. Please keep in mind that you will need to use process shared versions of these constructs.

The command line parameters and the format of the input file are the same as in Assignment 3:

Usage of your program:

$ transfProg inputFile numWorkers

So, as an example, assuming the input file is as below

1 1000
2 50
3 400
4 150
Transfer 1 2 200
Transfer 1 4 50
Transfer 2 3 100

your program should produce the following output regardless of the number of worker processes specified:

1 750
2 150
3 500
4 200

To get full credit, your solution should maximize concurrency, be free of race conditions and deadlocks, and produce the correct output. Please submit your files (source, README, and a Makefile) on CANVAS by the due date.