CS 374 Lab 5: Semaphore Practice

Each person in your group should clearly print their name here to so I know who worked in your group; I only need one hand-in from each group:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. Consider the fork-join implementation below. Draw the process flow graph for this, then implement it using semaphores inside a single cobegin-coend block. Components and labels for the drawing are provided to make this easier for your group.

T1 = 2; T2 = 3;

fork L1; fork L2; quit;

L1: p1; fork L3; quit;

L2: p2; fork L5; join T1, L6; quit;

L3: p3; fork L4; join T2, L7; quit;

L4: p4; join T1, L6; quit;

L5: p5; join T2, L7; quit;

L6: p6; join T2, L7; quit;

L7: p7; …

p1

p4

p5

p3

p2

p7

p6

ε

ε

Put your implementation using semaphores here:

semaphore S1 = 0; S2 = 0; …

cobegin

Xxx

//

Yyy

coend;

1. Consider the following implementation with semaphores.

semaphore S1 = 0; S2 = 0; S3 = 0; S4 = 0; S5 = 0;

cobegin

p1; V(S1);

//

p2; V(S1);

//

P(S1); P(S1); V(S2); p3; V(S3); V(S3);

//

P(S2); p4; V(S4); V(S4);

//

P(S3); p5; V(S5);

//

P(S3); p6; V(S5);

//

P(S5); P(S5); p7;

//

P(S4); p8;

//

P(S4); p9;

coend;

* 1. Rewrite this into S&P notation.
  2. Convert this into nested cobegin/coend blocks (no semaphores).

cobegin

Xxx

//

Yyy

coend;

**Check your answers with my key and get checked off when you’re finished!**