Assignment 1

# CS 374 – Operating Systems

## DUE: Tuesday, September 27

**Please type up answers to the following problems. An upload link will be provided on Canvas.**

1. Problem 1 in Bic & Shaw, page 66.
2. Problem 2 in Bic & Shaw, page 67.
3. Problem 3 in Bic & Shaw, page 67.
4. Problem 6 in Bic & Shaw, page 67.
5. Consider the following precedence graph:

p5

p1

p8

p2

p3

p7

p4

p9

p6

* 1. Express this graph using *fork*, *join*, and *quit* primitives
  2. Express this graph using a **single** cobegin-coend block with semaphores to control the precedence

1. Refer to the code segment below. It might be helpful to think of the expressions as comprising large matrix operations. Note that operations are frequently dependent on the completion of previous operations: for example, Q1 cannot be calculated until M2 has been calculated.
   1. Express the code as a process flow graph maintaining the expressed precedence of operations (eg: M1 must be calculated before QR2 is calculated). Use the **left hand sides** of the equation to label processes in your process flow graph. NOTE: part a) is a bit tricky—you will need to use some empty (or *epsilon* *transition*) arcs—that is, arcs not labeled by processes—to get the best graph. You will likely have crossed arcs (try to minimize this by redrawing as necessary).
   2. Implement the process flow graph using fork, join, and quit. Ensure that the maximum parallelism is achieved in both parts of this problem! If the graph from the first part is correct, this part is actually easy.

**M1** = A1 \* A2;

**M2** = (A1+A2)\*B1;

**QR2** = M1\*A1;

**Q1** = M2 + B2;

**QR1** = M2-M1;

**QR3** = A1\*B1;

**Z1**=QR3-QR1;