**Review Lab Exercises**

**Today’s class is a review Lab. We have covered the Labs on the following OS concepts: (a) Processes, (b) Threads (c) Signals.**

**For the exercise in the lab on Processes, please solve it.**

**For those who have solved, please ensure that it works correctly before building the project upon it [the project is an extension of the Lab 04 exercise]**

**For the other two labs, solve the following two problems as practice exercises.**

**Exercise 0)** [*Project: Processes*] Please view the project document and build upon Exercise of Lab 04. In particular, for today’s lab include **mycat** which you already did in Lab 03 in the Exercise of Lab04 as a starting point for your project..

**Exercise 1**) [*Threads*] Write a program to find the matrix multiplication C = A x B where A, B and C are 2-D dimensional arrays of type int and of size N (assume max value for N is 5). The main thread should initialize the matrices A and B with integers between 0 and 10 [assigned randomly], then creates N x N (N2) threads such that each thread calculates

C[i][j] = SUM(A[i][k] \* B[k][j]), for example

**C[1][1] = A[1][1]\*B[1][1] + A[1][2]\*B[2][1] + A[1][3]\*B[3][1] + A[1][4]\*B[4][1] + A[1][5]\*B[5][1]**

Finally, the main thread prints the three arrays A, B, and C as matrices

**Exercise 2**) [*Signals and Processes*] The SIGSTOP and SIGCONT signals suspend and resume a process, respectively. They are used by the Linux shells to support job control to implement built-in commands like stop, fg, and bg.

Write a main program that creates two children that both enter an infinite loop and display a message every second. The main program waits for three seconds and then suspends the first child. The second child continues to execute as usual. After another three seconds, the parent restarts the first child, waits a little while longer (3 seconds), and then terminates both children.