CS342 Operating Systems - Spring 2017
Project 3: Synchronization, Mutex and Condition Variables

Assigned: Mar 24 , 2017
Due date: Apr 06, 2017, 23:55

Objective: *Practicing synchronization, mutex and condition variables, multithreaded programming with Pthreads (POSIX threads), performing experiments and collecting measurement data, applying Probability and Statistics knowledge.*

In this project, you will write a program that will implement a deadlock-free solution with maximum concurrency for the dining philosopher problem. The program will work for any odd number of philosophers. The maximum number of philosophers can be 27. Each philosopher will be represented by a thread in your program. The main thread will create these threads. The main thread will not be a philosopher. You will use Pthreads library. You will use Pthread mutex and condition variables to synchonize. You can adapt the monitor based solution given in the textbook, that is deadlock free. There are also solutions using semaphores that are deadlock free. You can adapt such a solution as well. The program will be called `phil` and will take the following arguments.

phil <numphil> <minthink> <maxthink> <mineat> <maxeat> <dist> <count>

Here, <minthink> is the minimum think time for a philosopher, <maxthink> is the maximum think time; <mineat> is the minimum eating time, <maxeat> is the maximum eating time. The unit is milisecond. The max value of these parameters can be 60 seconds. The minimum value of these parameters can be 1 ms. <dist> can be one of "uniform", "exponential". The mean parameters for exponential distributions will be (minthink+maxthink)/2 or (mineat+maxeat)/2. The think time and eat time for a philosopher will be selected randomly according to the specified distribution and the related mean value. For exponential distribution think simple here: generate a random variate (value) with exponential distribution; and if the variate is not in the range, repeat (till you generate a value in range). Each philosopher will eat <count> times. After all philosophers eat count times, the program will terminate.

An example invocation of the program can be:

phil 5 500 1000 50 100 exponential 100

Report: Write a report that describes your solution. Write also how you generated the random values. Try to measure the duration of HUNGRY state for each philosopher. At the end, for each philosopher, find out the avg hungry duration and the standard deviation of hungry duration.

You will **submit** through Moodle. Submit: Makefile, phil.c, README.txt, report.pdf. Submission procedure is the same with the previous projects.

Clarifications:

- You need to learn how to use Pthreads mutex and condition variables. There are links to some resources in the References section of the course webpage. You can find additional resources from Internet.
- The project will be done individually.