## Operating Systems 2019 - 2020 Fall Project 2

Version 0.1

**Date**: 28/12/2019

**Due**: 09/01/2020 23:59

For any questions, contact TA, Osman Gökhan Uyan <gokhan.uyan at agu.edu.tr>.

Gains: Working on synchronization, mutex and condition variables, multithreaded

programming with Pthreads library (POSIX threads), reviewing your knowledge about probability and statistics.

In this project, you are supposed to write a program for the dining philosopher problem, which will implement a deadlock-free solution with maximum concurrency. The program should work for any number of philosophers (odd numbers, maximum 27). In the program, a thread will express a philosopher. These threads will be spawned by the main thread, which is not a philosopher. You need to use Pthreads, mutex and condition variables to synchronize. You can use monitor based solutions or semaphores that are deadlock free. The program will be named phsp and will take the following arguments: phsp <num\_phsp> <min\_think> <max\_think> <min\_dine> <max\_dine> <dst> <num>

For the arguments; <min\_think> is the minimum thinking time for a philosopher, <max\_think> is the maximum thinking time; <min\_dine> is the minimum dining time, <max\_dine> is the maximum dining time. Use miliseconds as unit. The highest value that these parameters can take is 60seconds and lowest value that these parameters can take is 1ms. <dst> is the distribution and it can be either “uniform” or “exponential”. The mean parameters for exponential distributions are (min\_think+max\_think)/2 or (min\_dine+max\_dine)/2. The thinking time and dining time for a philosopher will be selected randomly according to the specified dst and the mean value. For exponential distribution; simply generate a random value with exponential distribution; and if the value is not in range, repeat it. Each philosopher will dine <num> times. The program should terminate after all of the philosophers complete dining (num times).

A sample initiation could be:

phsp 5 500 1000 50 100 exponential 100

Report: Describe your solution. Write how you generated the random values. Try to measure the duration of hungry state for each philosopher, find out the average hungry state and the standard deviation of hungry state.

Submit: Makefile, phsp.c, README.txt, report.pdf.

Clarifications:

• You need to learn how to use Pthreads mutex and condition variables.

• The project must be done individually.