



ECE 316 - Operating Systems and Networking Laboratory

Practical Assignment 10 (Due: 01/04/2020)

Deliverables: Your report and code should be sent via email to lab teaching assistants (aanast01@ucy.ac.cy or ahadji29@ucy.ac.cy) prior the assignment examination and must include the usual cover page. In your report, include only the pseudocode, not the actual code, with any comments and description you may need to add, as well as a typical scenario that you use to test your programs. Email subject line should only consist of "ECE316_2020_10". Naming format for the zip file: lastName.zip (or lastName_lastName.zip if a group). **Caution:** Remove the executables (.exe) from the files before you send them!

[100%] Dining Philosophers Problem: There are **N philosophers** seated around a circular table and each philosopher can either think or eat. All philosophers are silent and do not communicate or coordinate with each other. On the table, there are **N plates of spaghetti** each one in front of each philosopher and **N chopsticks**, each one placed between two philosophers.

- a. Every philosopher thinks for some time and then he gets hungry and wants to eat.
- b. A philosopher must pick up its **two nearest** chopsticks in order to eat.
- c. A philosopher **can eat only when he holds two chopsticks**.
- d. A philosopher must pick up first one chopstick, then the second one, but **he cannot pick both at once**.
- e. Once the philosopher eats a little, he releases the chopsticks **one at a time** and starts thinking again.
- f. It is assumed that there is **infinite** supply of spaghetti.

Suppose that a philosopher just got hungry. First, he should pick one chopstick (if it is available) then he will pick the other chopstick (also if available) then he should eat for some time and finally he should release the chopsticks one at a time.

Write a multi-threaded program using Pthreads that takes as input the number **N** of philosophers and allocates these limited resources (chopsticks) among several processes (philosophers) in a manner that it is not possible to arrive at a state where each philosopher is waiting for the others to finish but nobody can eat at the end (deadlock) or a state where some philosophers will starve because they not manage to get both chopsticks.

Note: Your solution should use **Condition-Variables/Monitors** (`pthread_cond`) and **Mutex** (`pthread_mutex`)