Operating Systems
HCMC International University

Course: IT077IU
Time: 120 minutes

Laboratory Session 5

This session aims at revising concurrent processes/threads. The focus is on implementing programs to do some particular tasks using thread synchronization.

Problem 5.1: *Dining Philosophers Problem* (10 points)

Write a C program `dpp` that solves the dining philosophers problem. The program must satisfy requirements:

- Number of philosophers $N > 3$ is taken from the input
- Number of food plates is 2
- No two philosophers can use the same fork at the same time (mutual exclusion)
- No two philosophers eat the same food plates the same time (mutual exclusion)
- No philosopher must starve to death (avoid deadlock and starvation)

The program must handle error situations (including wrong input) in a meaningful way. Make sure the program compiles cleanly with gcc -O2 -Wall -lm -lpthread. You can refer to the algorithm in Chapter 6

The solution (only one .c text file) is formatted in *name_id_l5.c, no space* and submitted to the Blackboard system by the end of the lab class. Note that students are responsible for missing/duplicated files due to wrong formats/behaviors. Copying the whole source code from various sources such as the Internet is disallowed.