CS342 Operating Systems - Fall 2021
Project #3 – Synchronization

Assigned: Nov 22, 2021, 2021.
Due date: Dec 10, 2021, 23:59. Document version: 1.2

- *The project can be done in groups of 2 or individually. It will be done in C.*

Assignment

In this project you will implement the dining philosophers problem's solution (the monitor solution given in the textbook) using *Pthreads mutex and condition variables*. There will be 5 philosophers, numbered from 0 to 4. Neighboring philosophers should not eat at the same time. You program will create a separate thread for each philosopher. Each such thread will simulate the behaviour of a philosopher. Hence besides the main thread, there will be 5 other threads in your program.

You have to use mutex and condition variables. You can not use semaphores or any other primitive. You can not use monitor, since there is no monitor construct in C.

A philosopher can eat for a random time (uniform) between 1 and 5 seconds and think for a random time (uniform) between 1 and 10 seconds. Each philosoher is simulated by a thread. Hence the thread will run forever (in an endless while loop).

You will not use the same random vaue for all threads. Please use different seeds in generating random values in each thread. That means each thread will have a different seed used to generate random values.

The program will be called as phil (source phil.c). An example invocation can be:

./phil

For example, it will print out about who is eating like the following:

```
philosopher 3 started eating now.
philosopher 1 started eating now.
philosopher 3 finished eating now.
philosopher 1 finished eating now.
```

Submission

Put your phil.c file and all other files (including a Makefile) into a directory named with your ID (one of the students' ID is enough). Then tar and gzip the directory. For example, a student (of a group) with ID 21404312 will create a

directory named “21404312” and will put the files there. Then he/she will tar the directory (package the directory) as follows:

```
tar cvf 21404312.tar 21404312
```

Then he/she will gzip the tar file as follows:

```
gzip 21404312.tar
```

In this way he/she will obtain a file called 21404312.tar.gz. Then he/she will upload this file into Moodle.