Operating Systems
HCMC International University

Course: IT077IU
Time: 120 minutes

Laboratory Session 3

This session aims at revising POSIX threads programming on Linux. The focus is on implementing programs to do some particular tasks using multiple threads.

Problem 2.1: *POSIX threads programming on Linux* (0 points)

POSIX threads library supports several pthread APIs used to create and control threads.

- Create a new thread

```
#include <pthread.h>
int pthread_create(pthread_t *thread, const pthread_attr_t *attr,
                   void *(*start_routine) (void *), void *arg);
```

- Join with a terminated thread

```
#include <pthread.h>
int pthread_join(pthread_t thread, void **retval);
```

- Terminate calling thread

```
#include <pthread.h>
void pthread_exit(void *retval);
```

On Linux's terminal, try command *man* [*function name*] to refer the manual page of the function.

Problem 2.2: *Multi-thread Fibonacci program* (10 points)

Write a C/C++ program that accepts positive integral numbers n from the command line and verifies whether those numbers are members of the Fibonacci chain. The program contains multiple threads and each thread verifies a number with the answer at the end. An execution of the program on the command line might look like this:

```
$ mtfibo 12 3 19 4
4 is not a Fibonacci member
3 is a Fibonacci member
19 is not a Fibonacci member
12 is not a Fibonacci member
```

The program must handle error situations (including wrong input) in a meaningful way. Make sure the program compiles cleanly with gcc -O2 -Wall -lm -pthread.

The solution (only one .c text file) is formatted in *name_id_l3.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.