

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 Growing program* (10 points)

Write a C/C++ program that accepts N positive integral numbers from the command line and verifies whether those numbers are growing numbers. A growing number contains digits (from left to right) arranging in an increasing order, e.g., 1245, 258, ... At the end, the program contains multiple threads and each thread verifies a number with the answer. An execution of the program on the command line might look like this:

```
$ grow 1 23 144 3689
1 is not a growing number
23 is a growing number
144 is not a growing number
3689 is a growing number
```

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 Black-board 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.