

Laboratory Session 4

This session aims at revising processes and threads synchronization. The focus is on implementing programs to do some particular tasks using POSIX semaphore library.

Problem 4.1: *POSIX semaphore library on Linux* (0 points)

POSIX semaphore library supports several semaphore APIs used to control resources shared between threads.

- Initialize an unnamed semaphore

```
#include <semaphore.h>
int sem_init(sem_t *sem, int pshared, unsigned int value);
```

- Lock a semaphore

```
#include <semaphore.h>
int sem_wait(sem_t *sem);
```

- Unlock a semaphore

```
#include <semaphore.h>
int sem_post(sem_t *sem);
```

- Destroy an unnamed semaphore

```
#include <semaphore.h>
int sem_destroy(sem_t *sem);
```

Problem 4.2: *Producer and consumer problem* (10 points)

Write a C program `p2cp` that solves the producer and consumer problem. The program must satisfy requirements:

- Two producers generate integers and places them in an array
- Two consumers take integers out of the array one at time
- Only the producers or the consumers may access the array at any one time
- The producers cannot add integers into the full array and the consumers cannot remove integers from the empty array
- The array is a bounded buffer

You can refer to the bounded buffer algorithm in Chapter 5. 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`.

The solution (only one .c text file) is formatted in *name_id_l4.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.