

EMLab1 Report

學生：林咏毅

學號：b10901059

Review Question

1. What is the effect or the meaning of the initial value of the semaphore?

Q: To specify whether variables in the current process can be used in other process and child process.

2. What is the purpose of the C keyword volatile in the program?

Q: To tell the compiler that it need to repeatedly read the value of the variable rather than directly read from the cache.

3. What are the roles played by the parameters of methods fall() and rise() in class InterruptIn?

Q: They are the callback function called when the falling and rising edges occurred.

Discussions of Work

```
1  #define LED_DELAY 1000ms
2
3  void led_thread(void const *name) {
4      while (1) {
5          led_sem.acquire();
6          while (1) {
7              if (*((int*)name) == 2) {
8                  LD2_TOG;
9                  ThisThread::sleep_for(LED_DELAY);
10                 printf("led2\n");
11                 if(button_switch % 2 == 1)
12                     break;
13             }
14             else if (*((int*)name) == 1) {
15                 LD1_TOG;
16                 ThisThread::sleep_for(LED_DELAY);
17                 printf("led1\n");
18                 if (button_switch % 2 == 0)
19                     break;
20             }
21         }
22         LD1_OFF;
23         LD2_OFF;
24         LD3_OFF;
25         LD4_OFF;
26         led_sem.release();
27     }
28 }
```

```

1  int main()
2  {
3      LD1_OFF;
4      LD2_OFF;
5      LD3_OFF;
6      LD4_OFF;
7
8      led_sem.release();
9      button.fall(&button_pressed);
10     button.rise(&button_released); // switch led
11     const int a2 = 2;
12     const int a1 = 1;
13     t2.start(callback(led_thread, (void *)&a2));
14     t3.start(callback(led_thread, (void *)&a1));
15     while (1);
16 }

```

1. By simply modifying the LED_DELAY to 1000ms, we can achieve the first function.
2. Adding led_sem.release() to main() and modifying the led3 to led1 in led_thread() to achieve the second function.