실습 (me-thread): Mutual Exclusive - Thread

아래 프로그램 me-thread.c을 컴파일하고 실행하시오.

컴파일 방법

gcc -o me-thread me-thread.c -lpthread

실행

./me-thread &

- 1. 실행하면 몇 개의 스레드가 수행되는가?
- 2. 각 스레드에서 "gvar = gvar + 1; " 은 몇 번씩 수행되는가? (결국 각 스레드별로 gvar 값을 얼마를 증가시키게 되는가?)
- 3. 모든 스레드가 증가시킨 총 합은 얼마인가? 프로그램이 마지막으로 출력한 gvar 값과 일치하는가?
- 4. 앞서 실습한 race-thread 프로그램과 비교해보시오. 어떤 부분이 달라졌는가?
- 5. 프로그램에서 Critical region에 해당하는 부분을 적으시오. Critical region에 한 순간에 1개의 스레드만 수행될 수 있게 하기 위해 어떤 것(함수)을 사용하고 있는가?

[me-thread.c]

```
#define ITER (5000000)
#include <pthread.h>
#include <stdio.h>
int gvar = 0;
void * X(void *i);
pthread_mutex_t m;
int main(void)
{
      pthread_t t1;
      pthread_attr_t attr;
      void *status;
      int j;
      pthread_attr_init(&attr);
      pthread_attr_setscope(&attr, PTHREAD_SCOPE_SYSTEM);
      pthread_create(&t1, &attr, X, NULL);
      // Main thread
      for (j = 0; j < ITER; j++) {
            pthread_mutex_lock(&m);
             gvar = gvar + 1;
            pthread_mutex_unlock(&m);
      pthread_join(t1, &status);
      printf("%d\n", gvar);
void * X(void *i)
      int j;
      // Child thread
      for (j = 0; j < ITER; j++) {
            pthread_mutex_lock(&m);
             gvar = gvar + 1;
            pthread_mutex_unlock(&m);
      }
      pthread_exit(NULL);
}
```