Mutex (MUTual EXclusion)

Multicore Programming

Introduction

- What is Mutex?
- Pthread Mutex API
- Example

What is Mutex?

- Concurrent programming에서 공유 자원의 독점적 사용을 위한 메커니즘
- Atomicity, Singularity, Non-Busy Wait

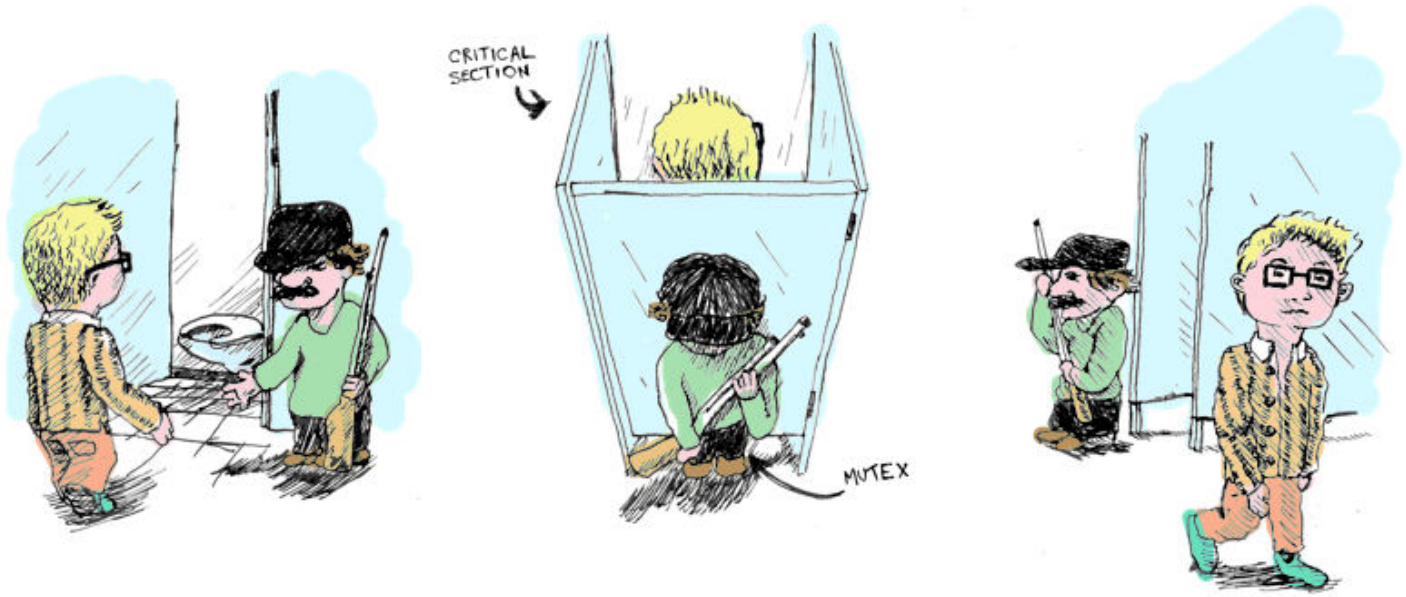

사진 출처: http://www.rudyhuyn.com/blog/2015/12/31/synchroniser-ses-agents-avec-lapplication/mutex/

Pthread Mutex API

- pthread_mutex_init
- pthread_mutex_lock
- pthread_mutex_unlock
- more APIs, but not today

Pthread Mutex API – pthread_mutex_init

```
int pthread_mutex_init(pthread_mutex_t *mutex,
                       const pthread_mutexattr_t *mutexattr);
```

- Mutex 객체를 초기화한다.

@param [in]	mutex	초기화할 Mutex 객체
@param [in]	mutexattr	Mutex의 attribute 설정할 때 사용.(e.g.,Deadlock Checking). 기본값 0.
@return		항상 return 0.

Pthread Mutex API – pthread_mutex_lock

```
int pthread_mutex_lock(pthread_mutex_t *mutex);
```

- Mutex 객체를 잠근다. 이미 잠겨있는 경우 사용가능할 때 까지 Block.

@param [in]	mutex	잠그려는 Mutex 객체
@return		성공하면(acquired) 0, 실패하면 mutexattr에 따른 에러값.

Pthread Mutex API – pthread_mutex_trylock

```
int pthread_mutex_trylock(pthread_mutex_t *mutex);
```

- Mutex 객체를 잠근다. 이미 잠겨있는 경우 즉시 return.

@param [in]	mutex	잠그려는 Mutex 객체
@return		성공하면(acquired) 0, 실패하면 mutexattr에 따른 에러값.

Pthread Mutex API – pthread_mutex_unlock

```
int pthread_mutex_unlock(pthread_mutex_t *mutex);
```

- Mutex 객체의 잠금을 해제한다.

@param [in]	mutex	잠금 해제하려는 Mutex 객체
@return		성공하면(released) 0, 실패하면 mutexattr에 따른 에러값.

Example

< prac_mutex.cpp >

```cpp
#include <stdio.h>
#include <pthread.h>

#define NUM_THREAD      10
#define NUM_INCREASE    1000000

int g_cnt_global = 0;
pthread_mutex_t g_mutex = PTHREAD_MUTEX_INITIALIZER;

void *ThreadFunc(void *arg) {
    long cnt_local = 0;

    for (int i = 0; i < NUM_INCREASE; i++) {
        pthread_mutex_lock(&g_mutex);
        g_cnt_global++;
        pthread_mutex_unlock(&g_mutex);
        cnt_local++;
    }

    return (void*)cnt_local;
}
```

Example (continue..)

```
23 int main(void) {
24     pthread_t threads[NUM_THREAD];
25
26     for (int i = 0; i < NUM_THREAD; i++) {
27         if (pthread_create(&threads[i], 0, ThreadFunc, NULL) < 0) {
28             return 0;
29         }
30     }
31
32     long ret;
33     for (int i = 0; i < NUM_THREAD; i++) {
34         pthread_join(threads[i], (void**)&ret);
35         printf("thread %d, local count: %d\n", threads[i], ret);
36     }
37
38     printf("global count: %d\n", g_cnt_global);
39
40     return 0;
41 }
```

Example (continue..)

< Result >

```
mrbin2002@ubuntu:~/TA_multicore/prac_mutex$ time ./a.out
thread 764360448, local count: 1000000
thread 755967744, local count: 1000000
thread 747575040, local count: 1000000
thread 739182336, local count: 1000000
thread 730789632, local count: 1000000
thread 722396928, local count: 1000000
thread 714004224, local count: 1000000
thread 705611520, local count: 1000000
thread 697218816, local count: 1000000
thread 688826112, local count: 1000000
global count: 10000000

real    0m1.493s
user    0m0.748s
sys     0m9.584s
```

Example (continue..)

< g_cnt_global++에 해당하는 assembly instruction >

```
32      movl    $g_mutex, %edi
33      call    pthread_mutex_lock
34      movl    g_cnt_global(%rip), %eax
35      addl    $1, %eax
36      movl    %eax, g_cnt_global(%rip)
37      movl    $g_mutex, %edi
38      call    pthread_mutex_unlock
39      addq    $1, -8(%rbp)
40      addl    $1, -12(%rbp)
```

Critical section

Thank You