Thread & Mutex

System Programming

2019 여름 계절학기

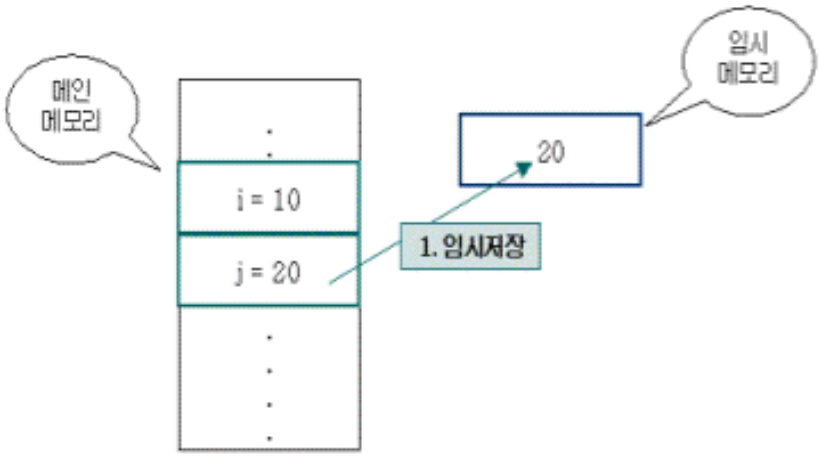
한양대학교 공과대학 컴퓨터소프트웨어학부
홍석준

임계 영역

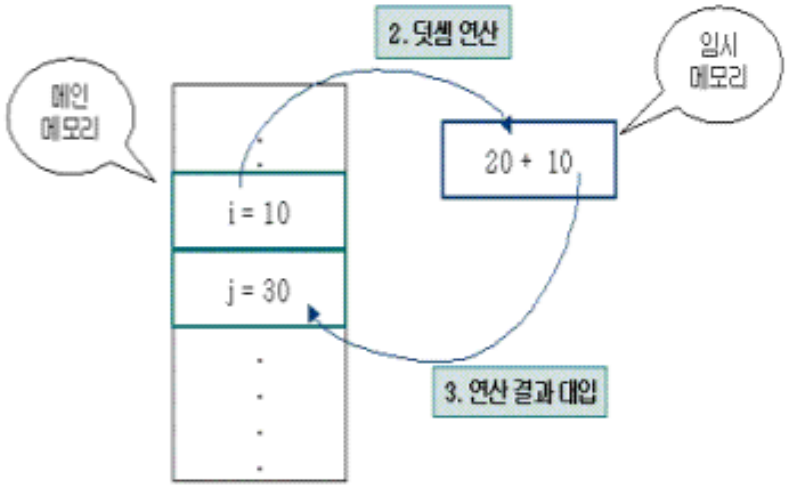
임계 영역(critical section)

❑ 컴퓨터가 덧셈을 하는 기본 원리

```
int i = 10
int j = 20
j+=i
```



[그림 17-8]

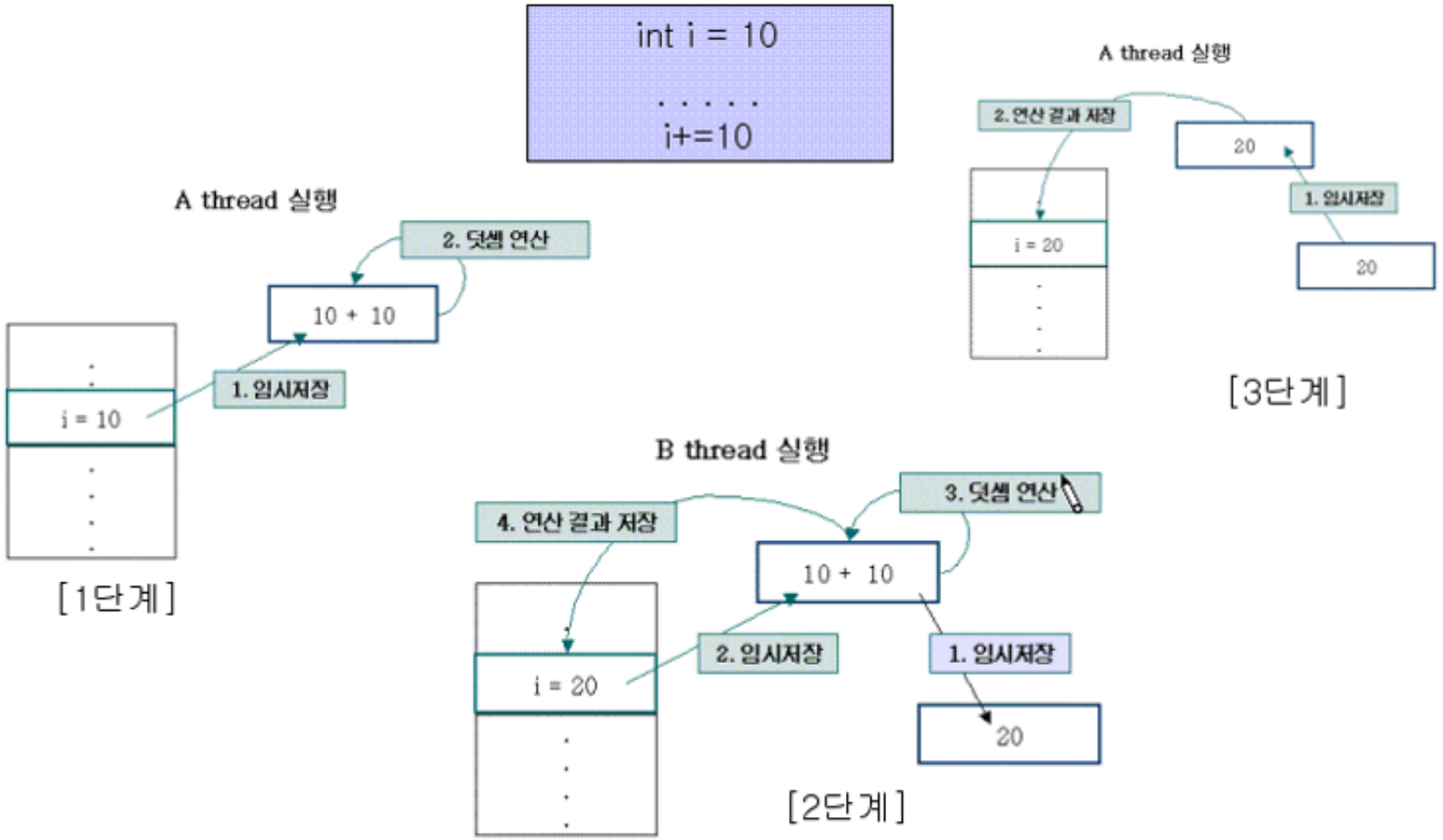


[그림 17-9]

임계 영역

임계 영역(critical section)

❑ 두 개의 쓰레드에 의한 덧셈 연산



동기화 (synchronization)

1. 임계 영역(critical section)

- 둘 이상의 쓰레드에 의해서 공유되는 메모리 공간에 접근하는 코드 영역

2. 동기화

- 첫째 : 공유된 메모리에 둘 이상의 쓰레드가 동시 접근하는 것을 막는 행위
- 둘째 : 둘 이상의 쓰레드 실행 순서를 컨트롤 하는 행위

3. 대표적인 동기화 기법

- 뮤텍스, 세마포어

뮤텍스

1. 무엇을 의미하는 것인가?

- pthread_mutex_t 타입의 변수를 가리켜 뮤텍스라고 표현한다.
- 일종의 문고리에 해당한다.

2. 기본 원리는 무엇인가?

- 임계 영역에 들어갈때 뮤텍스(문고리)를 잠그고 들어간다.
- 임계 영역에서 빠져 나올 때 뮤텍스를 풀고 나간다.

3. 뮤텍스를 조작하는 함수들 (모두 pthread.h에 정의되어있음)

- 뮤텍스 초기화 함수 : pthread_mutex_init
- 뮤텍스 소멸 함수 : pthread_mutex_destroy
- 뮤텍스 잠그는 함수 : pthread_mutex_lock
- 뮤텍스 풀어주는 함수 : pthread_mutex_unlock

뮤텍스

뮤텍스 함수

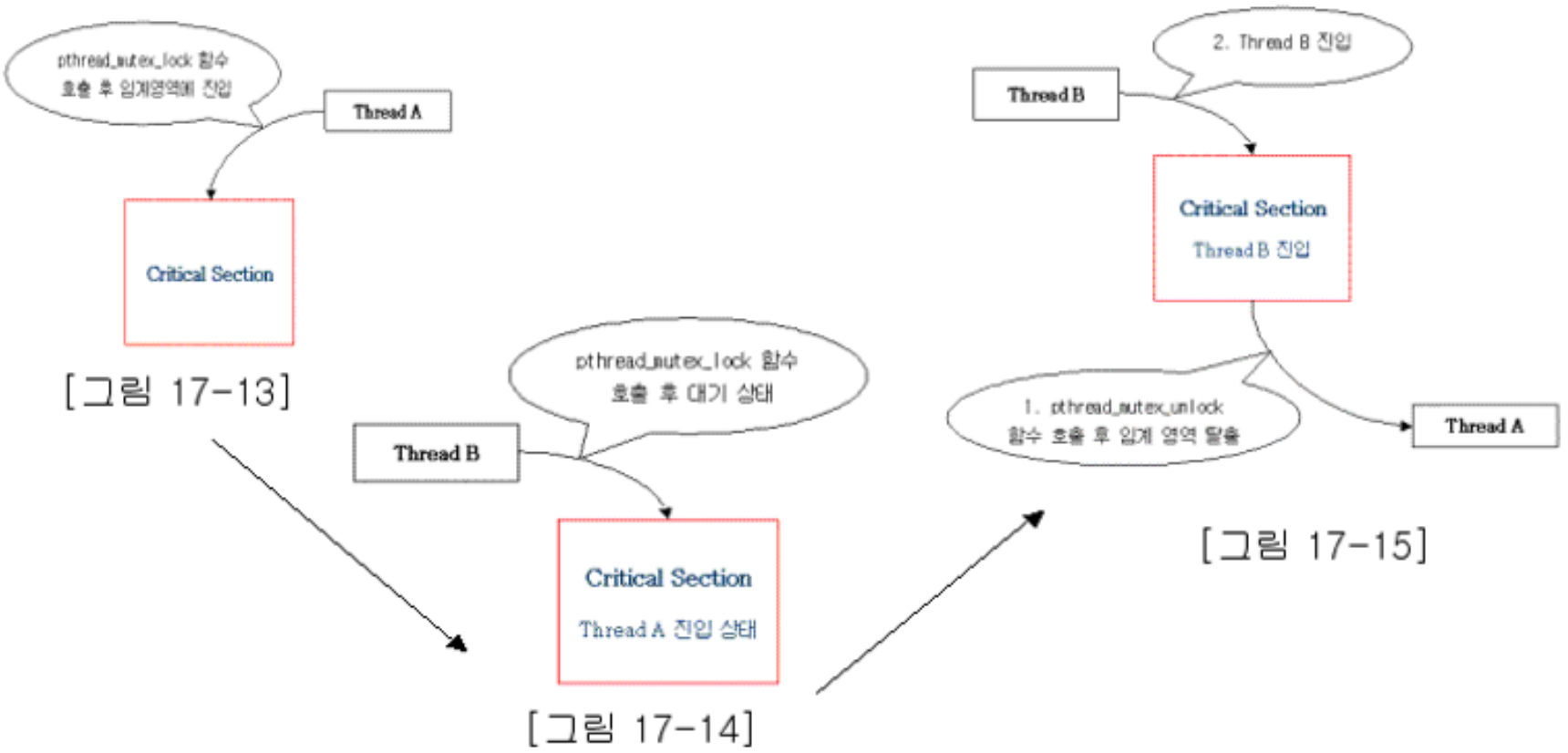
```
#include <pthread.h>

int pthread_mutex_init(pthread_mutex_t *restrict mutex,
                       const pthread_mutexattr_t *restrict attr);

int pthread_mutex_destroy(pthread_mutex_t *mutex);
```

Both return: 0 if OK, error number on failure

```
#include <pthread.h>

int pthread_mutex_lock(pthread_mutex_t *mutex);

int pthread_mutex_trylock(pthread_mutex_t *mutex);

int pthread_mutex_unlock(pthread_mutex_t *mutex);
```

All return: 0 if OK, error number on failure

뮤텍스 동기화 원리


pthread_mutex_lock 함수
호출 후 임계영역에 진입
Thread A
Critical Section
[그림 17-13]
pthread_mutex_lock 함수
호출 후 대기 상태
Thread B
Critical Section
Thread A 진입 상태
[그림 17-14]
2. Thread B 진입
Thread B
Critical Section
Thread B 진입
1. pthread_mutex_unlock
함수 호출 후 임계 영역 탈출
Thread A
[그림 17-15]

뮤텍스 예제 코드

❑ mutex.c 코드 작성

```c
#include <stdio.h>
#include <unistd.h>
#include <stdlib.h>
#include <pthread.h>

void * thread_increment (void *arg);
char thread1[] = "A Thread";
char thread2[] = "B Thread";

pthread_mutex_t mutx;
int number=0;

int main()
{
        pthread_t t1, t2;
        void *thread_result;
        int state;

        state = pthread_mutex_init(&mutx, NULL);
        if(state){
                puts("Mutex initializaiton fail");
                exit(1);
        }

        pthread_create(&t1, NULL, thread_increment, &thread1);
        pthread_create(&t2, NULL, thread_increment, &thread2);

        pthread_join(t1, &thread_result);
        pthread_join(t2, &thread_result);


        printf("Final number : %d \n", number);
        pthread_mutex_destroy(&mutx);


        return 0;
}
```

(계속)

뮤텍스 예제 코드

❑ mutex.c 코드 작성

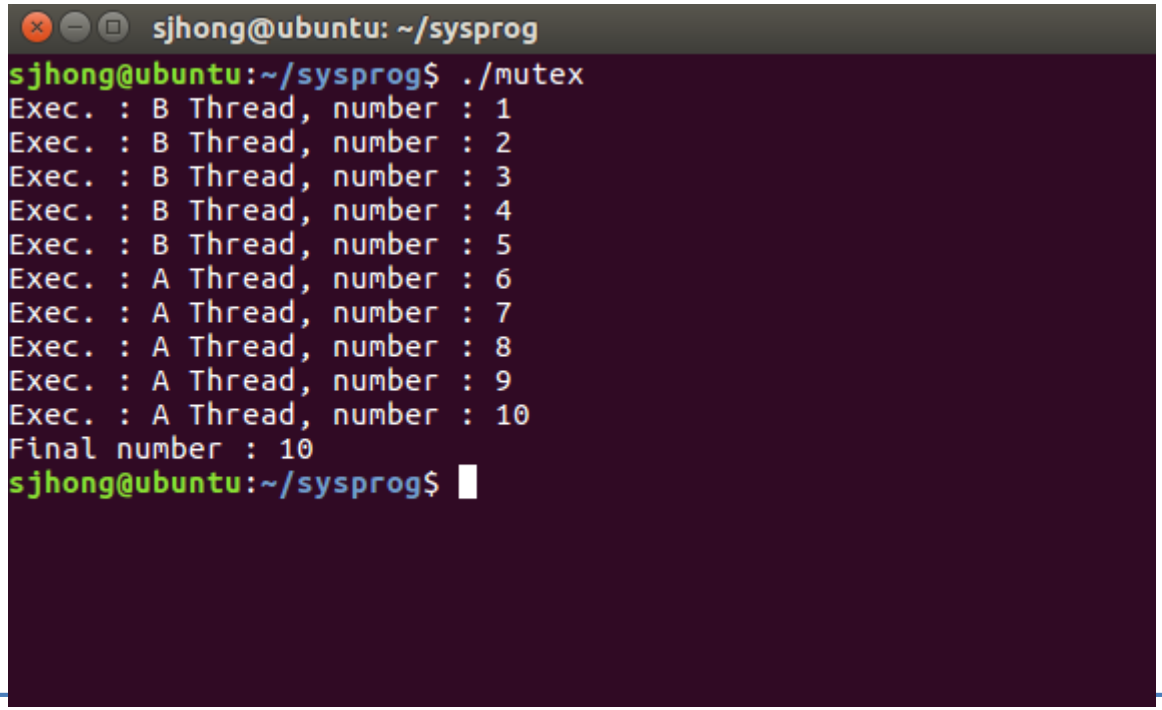
```
void * thread_increment(void * arg)
{
        int i;
        for(i=0;i<5;i++){
                pthread_mutex_lock(&mutx);
                sleep(1);
                number++;
                printf("Exec. : %s, number : %d \n", (char*)arg, number);
                pthread_mutex_unlock(&mutx);
        }
}
```

(끝)

뮤텍스 예제 코드

❑ mutex.c 실행 화면

- gcc –o mutex mutex.c -lpthread로 컴파일
- 두 개의 쓰레드가 출력
- 총 합 10이 출력



Q and A