

CS SCHOOL



Race condition

```
⊟#include <stdio.h>
 #include <Windows.h>
 #include cess.h>
 #define THREAD_NUM
                      30
 //전역변수, 즉 공유자원
 long long num = 0;
 //multithread main function
 unsigned WINAPI AddOne(void *);
 unsigned WINAPI SubtractOne(void *);
```



Race condition

```
for (int i = 0; i < THREAD_NUM; i += 2)
{
    threadArr[i] = (HANDLE)_beginthreadex(0, 0, AddOne, 0, 0, 0);
    threadArr[i + 1] = (HANDLE)_beginthreadex(0, 0, SubtractOne, 0, 0, 0);
}</pre>
```



Race condition

```
unsigned WINAPI AddOne(void *)
   for (int i = 0; i <= 100000; i++)
       num++;
    return 0;
unsigned WINAPI SubtractOne(void *)
   for (int i = 0; i <= 100000; i++)
       num--;
    return 0;
```



mutex

뮤텍스를 활용한 critical section 해결 방법

HANDLE hMutex;



공유 자원 권리

뮤텍스를 활용한 critical section 해결 방법

```
]unsigned WINAPI AddOne(void *)
h#if SWITCH
     WaitForSingleObject(hMutex, INFINITE);
#endif
     for (int i = 0; i <= 100000; i++)
         num++;
     cnt++;
]#if SWITCH
     ReleaseMutex(hMutex);
#endif
     return 0;
```



mutex

뮤텍스를 활용한 critical section 해결 방법

둘 이상의 thread가 공유 자원에 동시에 접근할 수 없도록 한 thread가 작업을 모두 수행하기 전에는 다른 thread가 접근할 수 없다.