Chapter 20. Windows에서의 쓰레드 동기화 내용 확인문제

1. 윈도우 운영체제의 유저모드, 커널모드와 관련해서 옳은 것을 모두 고르면?
   1. 유저모드는 응용 프로그램이 실행되는 기본모드로, 접근할 수 있는 메모리의 영역에는 제한이 없지만 물리적인 영역으로의 접근은 허용되지 않는다.
   2. 응용 프로그램이 실행되는 과정에서는 절대 커널모드로 진입하지 않는다. 응용 프로그램이 실행중인 과정에서는 유저모드로만 동작한다.
   3. 윈도우는 메모리의 효율적인 사용을 위해서 유저모드와 커널모드를 각각 별도로 정의하고 있다.
   4. 응용 프로그램이 실행되는 과정에서도 커널모드로의 변환이 발생할 수 있다. 단, 일단 커널모드로 변환이 되면, 프로세스는 이 상태로 실행을 계속 이어하게 된다.
2. 유저모드 동기화, 커널모드 동기화와 관련된 다음 문장들 중에서 말하는 바가 옳으면 O, 틀리면 X를 표시하자.
   * + 유저모드 동기화는 커널모드로의 전환을 수반하지 않는다. 즉 운영체제 레벨에서 제공되는 기능의 동기화가 아니다. ( O )
     + 커널모드 동기화는 운영체제를 통해서 제공되는 기능이므로, 유저모드 동기화에 비해서 많은 기능을 제공한다. ( O )
     + 커널모드 동기화 과정에서는 유저모드에서 커널모드로, 다시 커널모드에서 유저모드로의 전환과정이 수반된다는 단점이 있다. ( O )
     + 특별한 경우가 아니면 커널모드 동기화를 사용하는 것이 원칙이다. 유저모드 동기화는 커널모드 동기화가 제공되기 이전의 동기화 기법이다. ( X )
3. 본문의 예제 SyncSema\_win.c의 Read 함수는 임계영역을 빠져나가는데 오랜시간이 걸리도록 정의가 되어있다. 이에 대한 해결책을 제시하고 실제 예제에 적용해보자.

텍스트이(가) 표시된 사진

자동 생성된 설명

임계영역에서는 공유 데이터에 대한 접근을 수행한 후 최대한 빨리 빠져나와야 다른 프 로세스 또는 스레드의 접근이 용이하다. 이는 다중 스레딩 작업 환경에서 중요한 부분이다. 따라서 임계영역에서 대기시간이 발생할 수 있는 입력 함수를 임계 영역 바깥쪽에서 수행하도록 코드를 수정한다. 입력이 이루어지고, 임계 영역 진입에 대한 semaphore 검사가 이루어져야한다.

1. 본문의 예제 SyncEvent\_win.c를 세마포어 기반의 동기화 기법을 적용해서 동일한 실행결과를 보이도록 재 구현해 보자.

본문 예제는 Event 오브젝트 기반의 동기화 작업 코드 이다.

이를 Semaphore 오브젝트 기반의 동기화 작업으로 재 구현하여야 한다.

#include <stdio.h>

#include <windows.h>

#include <process.h>

#define STR\_LEN 100

unsigned WINAPI NumberOfA(void \* arg);

unsigned WINAPI NumberOfOthers(void \* arg);

static char str[STR\_LEN];

static HANDLE hsemaphore;

int main(int argc, char \*argv[])

{

HANDLE hThread1, hThread2;

hsemaphore = CreateSemaphore(NUll, 0, 2, NULL);

hThread1 = (HANDLE)\_beginthreadex(NULL, 0, NumberOfA, NULL, 0, NULL);

hThread2 = (HANDLE)\_beginthreadex(NULL, 0, NumberOfOthers, NULL, 0, NULL);

WaitForSingleObject(hThread1, INFINITE);

WaitForSingleObject(HThread2, INFINITE);

closeHandle(hsemaphore);

return 0;

}

unsinged WINAPI NumberOfA(void \* arg)

{

int i, cnt = 0;

WaitForSingleObject(hsemaphore, INFINITE);

for (i = 0; str[i]=0; i++)

{

if(str[i]=='A')

cnt++;

}

printf("Num of A: %d \n", cnt);

return 0;

}

unsinged WINAPI NumberOfOthers(void \* arg)

{

int i, cnt = 0;

WaitForSingleObject(hsemaphore, INFINITE);

for (i = 0; str[i]=0; i++)

{

if(str[i]!='A')

cnt++;

}

printf("Num of others: %d \n", cnt-1); // 개행문자 포함이 되기 때문에 1을 뺌.

return 0;

}