동기화:

특정 조건이 만족될 때까지 스레드를 대기시키는 것.

동기화의 목적:

데이터 보호. Mutex, Semaphore, Critical Section, SRW...

통지에 의한 흐름 제어. 소유권을 얻는 것도 어떤 사건이므로 위의 목적을 포함한다고 볼 수 있다.(서로 명확하게 구분될 필요는 없다.) 그 외에는 완료 통지. Event, Waitable Timer, IOCP

동기화의 방법:

커널 동기화 객체를 통한 Wating 상태로의 전환.

스핀 락 방식. 루프에서 조건 체크. 유저 동기화 객체에서 내부적으로 사용. 부하가 결국 많아지면 커널 모드와 혼합하기도 함.

기본적으로 유저 모드 객체가 커널 모드 보다 빠름.

커널 모드로의 전환이 최소화 되기 때문에 스레드 문맥 전환을 줄임.

동기화 전용 객체:

커널 - Mutex, Semaphore, Event, Waitable Timer...

유저 - Critical Section, SRW(Slim Reader-Writer)-Lock…

대기 함수

{

WaitForSingleObject/MultipleObjects

해당 커널 객체가 Signaled(보통 완료 되었을 때 혹은 동기화 객체를 쓸 수 있을 때) 상태가 될 때까지 대기.

return - WAIT\_OBJECT\_0(Multi: + nCount - 1), WAIT\_TIMEOUT, WAIT\_ABANDONED(Mutex. Multi: + nCount - 1), WAIT\_FAILED(GetLastError)

SignalObjectAndWait

Mutex, Semaphore, Event

WaitForInputIdle - IPC

MsgWaitForMultipleObjects - Message

WaitForDebugEvent - Debug App

비동기 입출력 관련.

등등.

}

Mutex(mutually exclusive):

독점적 소유권.

이 객체를 소유하는 스레드가 ReleaseMutex하지 않고 종료되면

대기 함수에서 WAIT\_ABANDONED가 리턴됨.

버려진 뮤텍스도 다른 대기 중인 스레드가 이용할 수 있음.

Semaphore.

동시에 여러 스레드가 소유 가능.

세마포어 응용 Thread Pool

스레드 메시지 큐

PostThreadMessage

GetMessage 메시지 루프

MsgWaitForMultipleObjects

PeekMessage