

Systemové programování Windows

Synchronizace vláken

Obsah

- ▶ **Uživatelský režim**
 - ▶ Interlocked...
 - ▶ CriticalSection

- ▶ **Objekty jádra**
 - ▶ Event
 - ▶ Mutex
 - ▶ Semaphore



Špatně

```
//  
//TAKTO NIKDY NE  
//  
void Lock( void )  
{  
    //cekani na zamek  
    while ( _locked )  
    {  
        Sleep(100);  
    }  
  
    //uzamceni  
    _locked = true;  
}
```



Interlocked...

```
LONG __cdecl InterlockedIncrement(  
    _Inout_ LONG volatile *Addend );
```

```
LONG __cdecl InterlockedDecrement(  
    _Inout_ LONG volatile *Addend );
```

```
LONG __cdecl InterlockedCompareExchange(  
    _Inout_ LONG volatile *Destination,  
    _In_ LONG Exchange,  
    _In_ LONG Comparand );
```



CriticalSection

```
VOID InitializeCriticalSection(PCRITICAL_SECTION cs);
```

```
VOID DeleteCriticalSection(PCRITICAL_SECTION cs);
```

```
VOID EnterCriticalSection(PCRITICAL_SECTION cs);
```

```
VOID LeaveCriticalSection(PCRITICAL_SECTION cs);
```



WaitForSingleObject

```
DWORD WaitForSingleObject(  
    HANDLE objectHandle,  
    DWORD milliseconds );
```

Návratové hodnoty:

- ▶ `WAIT_OBJECT_0`
- ▶ `WAIT_TIMEOUT`
- ▶ `WAIT_FAILED`



Event

```
HANDLE CreateEvent (  
    PSECURITY_ATTRIBUTES sa,  
    BOOL manualReset,  
    BOOL initialState, //true = signaled  
    PCTSTR name );
```

```
BOOL SetEvent (HANDLE eventHandle);
```

```
BOOL ResetEvent (HANDLE eventHandle);
```



Mutex

```
HANDLE CreateMutex (  
    PSECURITY_ATTRIBUTES sa,  
    BOOL initialOwner,  
    PCTSTR name );  
  
BOOL ReleaseMutex (HANDLE mutexHandle);
```



Semaphore

```
HANDLE CreateSemaphore (  
    PSECURITY_ATTRIBUTE sa,  
    LONG initialCount,  
    LONG maximumCount,  
    PCTSTR name );
```

```
BOOL ReleaseSemaphore (  
    HANDLE semaphoreHandle,  
    LONG releaseCount,  
    PLONG previousCount );
```



Díky za pozornost

