

Vlákná ve Windows

Vláknem je samostatná jednotka v procesu. Tradiční procesy běží v jednom vlákně. Úlohu lze rozdělit do více souběžně běžících samostatných procesů a umožnit tak souběžné vykonávání její částí, nicméně to má určité nevýhody, jako je problém sdílení společných zdrojů a vzájemná synchronizace procesů. Efektivnější a jednodušší je mít jen jeden proces a v něm více vláken.

Vlastnosti vláken

- sdílí stejný virtuální adresní prostor
- sdílí stejnou globální paměť procesu
- každé vlákno má svůj vlastní zásobník

Vytvoření vlákna

```
HANDLE WINAPI CreateThread(  
    _In_opt_ LPSECURITY_ATTRIBUTES lpThreadAttributes,  
    _In_     SIZE_T dwStackSize,  
    _In_     LPTHREAD_START_ROUTINE lpStartAddress,  
    _In_opt_ LPVOID lpParameter,  
    _In_     DWORD dwCreationFlags,  
    _Out_opt_ LPDWORD lpThreadId  
);
```

Parametry funkce:

lpThreadAttributes	NULL (implicitní hodnota <i>security</i> parametru)
dwStackSize	0 (implicitní velikost zásobníku vlákna)
lpStartAddress	Ukazatel na funkci, která bude vykonávána ve vlákně. Je zároveň adresou začátku vlákna.
lpParameter	Parametr funkce vykonávané ve vlákně. Jeho datový typ je ukazatel. Může být i použit pro zadání celočíselné hodnoty (ve funkci je pak zapotřebí příslušná konverze ukazatele na datový typ číselné hodnoty). Není-li parametr využit, uveďte se hodnota NULL.
dwCreationFlags	0 CREATE_SUSPENDED – vlákno je vytvořeno v pozastaveném stavu a nebude spuštěno, dokud nebude zavolána funkce ResumeThread .
lpThreadId	NULL (nepotřebujeme identifikátor vlákna)

Návratová hodnota funkce:

Funkce vrací *handle* na vytvořené vlákno, když vytvoření vlákna bylo úspěšné. Pokud vrátí NULL, lze kód chyby zjistit funkcí [GetLastError](#).

Prototyp funkce vykonávané ve vlákně:

```
DWORD WINAPI ThreadFunction(LPVOID lpParameter);
```

Ukončení vlákna

Lze běžným návratem z funkce vykonávané ve vlákně (příkazem **return** výraz) nebo funkcí:

```
VOID WINAPI ExitThread(_In_ DWORD dwExitCode);
```

Po ukončení vlákno bude stále existovat a hodnotu, kterou jeho funkce vrátila, lze zjistit funkcí:

```
BOOL WINAPI GetExitCodeThread(  
    _In_ HANDLE hThread,  
    _Out_ LPDWORD lpExitCode  
);
```

Parametry funkce:

hThread	<i>Handle</i> vlákna, který vrátila funkce vytvoření vlákna <code>CreateThread</code> .
lpExitCode	Ukazatel na proměnnou datového typu <code>DWORD</code> , do které bude uložena hodnota vrácená vláknem.

Vlákno zrušíme, když jeho *handle* uzavřeme funkcí [CloseHandle](#).

Zastavení a obnovení činnosti vlákna

```
DWORD WINAPI SuspendThread(_In_ HANDLE hThread);  
DWORD WINAPI ResumeThread(_In_ HANDLE hThread);
```

Funkce vrací počet zastavení vlákna (hodnotu čítače zastavení).

V případě chyby vrací hodnotu (`DWORD`)-1 (kód chyby lze zjistit funkcí [GetLastError](#)).

Čekání na ukončení vlákna

```
DWORD WINAPI WaitForSingleObject(  
    _In_ HANDLE hHandle,  
    _In_ DWORD dwMilliseconds  
);
```

Čekání na ukončení více vláken

Pro tuto funkci je zapotřebí připravit pole, ve kterém budou uloženy *handles* všech vláken, na které má funkce čekat.

```
DWORD WINAPI WaitForMultipleObjects(  
    _In_ DWORD nCount,  
    _In_ const HANDLE *lpHandles,  
    _In_ BOOL bWaitAll,  
    _In_ DWORD dwMilliseconds  
);
```

Parametry funkce:

nCount	Počet <i>handles</i> uložených v poli.
lpHandles	Pole <i>handles</i> vláken, na které má funkce čekat.
bWaitAll	TRUE – čeká na ukončení všech vláken. FALSE – funkce se ukončí, když se ukončí některé z vláken. Návratová hodnota v tomto případě identifikuje vlákno, které se ukončilo.
dwMilliseconds	Počet milisekund určující čas, dokdy funkce bude čekat. Pro neomezený čas čekání se uvede hodnota parametru INFINITE .

Návratová hodnota funkce:

WAIT_OBJECT_0 až WAIT_OBJECT_0 + <i>nCount</i> -1	Čekání bylo úspěšně ukončeno.
WAIT_TIMEOUT	Určený čas čekání uplynul a čekání nebylo úspěšné.
WAIT_FAILED	Funkce selhala. Bližší informace o chybě lze získat funkcí GetLastError .