

Wielowątkowość w COM

Marek Mittmann
Grudzień 2002

Wielowątkowość w Windows

- ◆ Wątek (*ang. thread*) to sekwencja wykonywana przez proces, każdy proces ma do wykonania jeden lub więcej wątków
- ◆ Tam, gdzie wątki korzystają ze współdzielonych danych, konieczna jest synchronizacja
- ◆ Do synchronizacji wykorzystujemy: zdarzenia, sekcje krytyczne, semaforey i wzajemne wykluczanie obiektów (*mutex*)

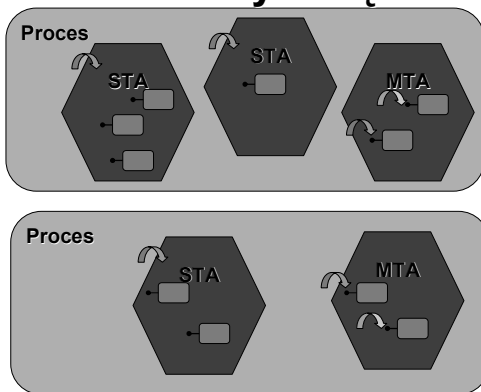
Przedziały

- ◆ Przedział (*ang. apartment*) to kolekcja obiektów wraz z ich kontekstami
- ◆ Każdy obiekt należy do dokładnie jednego przedziału
- ◆ Przedziały określają zestaw zasad dla danej grupy obiektów, wymaganych do poprawnej pracy w środowisku wielowątkowym

Typy przedziałów

- ◆ **Jednowątkowe (STA)**
 - Mają po jednym wątku, proces może mieć wiele STA
 - Synchronizowane przy pomocy kolejki komunikatów
- ◆ **Wielowątkowe (MTA)**
 - Mają wiele wątków, proces może mieć tylko jeden MTA
 - Brak synchronizacji
- ◆ **Neutralne (NA)**
 - Proces może mieć jeden NA

Przedziały a wątki



Przedziały jednowątkowe

- ◆ Każdy STA ma tylko jeden wątek, w pojedynczym procesie może istnieć wiele STA
- ◆ Wszystkie komponenty nie obsługujące modelu wielowątkowego działają w głównym STA
- ◆ Wątek deklaruje obsługę modelu STA przez wywołanie *CoInitializeEx* z drugim parametrem równym *COINIT_APARTMENTTHREAD*

Przedziały jednowątkowe c.d.

- ◆ Wywołania metod w STA są automatycznie synchronizowane i dysponowane przy użyciu kolejek komunikatów
- ◆ Każdy komponent zewnątrzprocesowy, który obsługuje model STA, musi zawierać pętlę komunikatów
- ◆ Komponent wykonywalny może blokować wywołania metod w STA za pomocą filtrów komunikatów (interfejs *IMessageFilter*)

Przedziały wielowątkowe

- ◆ W pojedynczym procesie może istnieć tylko jeden MTA, który z kolei może zawierać wiele wątków
- ◆ Wywołania metod w MTA nie są automatycznie synchronizowane
- ◆ Wątek deklaruje obsługę modelu STA przez wywołanie *CoInitializeEx* z drugim parametrem = *COINIT_MULTITHREAD*

Przedziały neutralne

- ◆ W pojedynczym procesie może istnieć tylko jeden NA
- ◆ Nie ma stałych wątków, obiekty z NA są zawsze wykonywane przez wątek wywołujący
- ◆ Brak wbudowanej synchronizacji
- ◆ Przedziały neutralne są obsługiwane tylko przez komponenty wewnątrzprocesowe

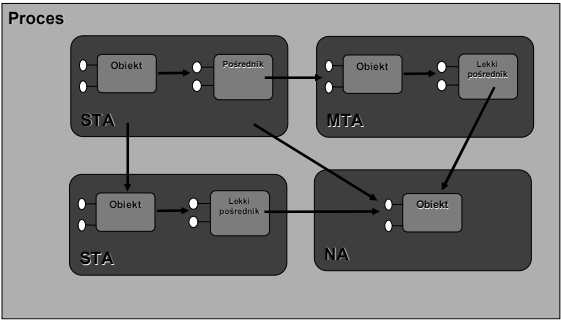
Przedziały a komponenty wewnątrzprocesowe

- ◆ Komponenty wewnątrzprocesowe nie wywołują *CoInitializeEx*, obsługiwany model wielowątkowości deklarują za pomocą wpisu w rejestrze (klucz *CLSID\InprocServer32*). Dopuszczalne wartości to:
 - *Apartment* – STA
 - *Neutral* – NA
 - *Free* – MTA
 - *Both* – obsługuje STA, NA i MTA
 - brak – komponent jednowątkowy

Przypisywanie do przedziałów

Typ przedziału twórcy	Model wielowątkowości komponentu				
	Nieokreślony	Apartment	Free	Both	Neutral
Główny STA	główny STA	główny STA	MTA	główny STA	NA
STA	główny STA	STA wywołującego	MTA	STA wywołującego	NA
MTA	główny STA	macierzysty STA	MTA	MTA	NA
NA (wątek STA)	główny STA	STA wywołującego	MTA	NA	NA
NA (wątek MTA)	główny STA	macierzysty STA	MTA	NA	NA

Współdziałanie przedziałów



Współdzielanie przedziałów

Typ przedziału twórcy	Model wielowątkowości komponentu				
	Nieokreślony	Apartment	Free	Both	Neutral
Główny STA	bezpośredni dostęp	bezpośredni dostęp	dostęp przez pośrednika	bezpśred ni dostęp	dostęp przez lekkiego pośrednika
STA	dostęp przez pośrednika	bezpośredni dostęp	dostęp przez pośrednika	bezpśred ni dostęp	dostęp przez lekkiego pośrednika
MTA	dostęp przez pośrednika	dostęp przez pośrednika	bezpśredni dostęp	bezpśred ni dostęp	dostęp przez lekkiego pośrednika
NA (wątek STA)	dostęp przez pośrednika	dostęp przez lekkiego pośrednika	dostęp przez pośrednika	bezpśred ni dostęp	bezpśredni dostęp
NA (wątek MTA)	dostęp przez pośrednika	dostęp przez pośrednika	dostęp przez lekkiego pośrednika	bezpśred ni dostęp	bezpśredni dostęp

Przekazywanie interfejsów pomiędzy przedziałami

```
// STA1
IMyInterface* pMyInterface;
CoCreateInstance(CLSID_MyCOMClass, NULL,
  CLSCTX_LOCAL_SERVER, IID_IMyInterface,
  (void**)&pMyInterface);

IStream* pStream;
CoMarshalInterThreadInterfaceInStream(
  IID_IMyInterface, pMyInterface,
  &pStream);

DWORD threadid;
CreateThread(0, 0,
  (LPTHREAD_START_ROUTINE) MyThread,
  (void*)pStream, 0, &threadid);

// ...
pMyInterface->Release();

void MyThread(IStream* pStream)
{
  // Utwórz STA2
  CoInitializeEx(NULL,
    COINIT_APARTMENTTHREADED);

  IMyInterface* pMyInterface;

  CoGetInterfaceAndReleaseStream(
    pStream, IID_IMyInterface,
    (void**)&pMyInterface);

  // ...

  pMyInterface->Release();

  CoUninitialize();
}
```

Podsumowanie

- ◆ Każdy STA może mieć tylko jeden wątek
- ◆ Komponenty wykorzystujące wątki STA muszą pobierać i dysponować komunikaty okna
- ◆ Przekazywanie wskaźników interfejsów pomiędzy wątkami z różnych przedziałów powinno odbywać się za pośrednictwem szeregowania
- ◆ Każdy wątek korzystający z COM powinien wywołać *CoInitializeEx*

Podsumowanie c.d.

- ◆ Każdy obiekt jest związany tylko z jednym, ale przedział może zawierać wiele obiektów
- ◆ Główny STA jest tworzony przez wątek, który pierwszy wywoła *CoInitializeEx* z *COINIT_APARTMENTTHREAD*
- ◆ Proces może mieć wiele STA, ale tylko jeden MTA i jeden NA
- ◆ Dla obiektów wewnątrzprocesowych należy zdefiniować w rejestrze *ThreadingModel*
