TWORZENIE SKŁADNIKÓW WEWNĄTRZPROCESOWYCH

Poniższy zwięzły opis przedstawia najważniejsze kroki niezbędne dla uzyskania klienta i serwera wewnątrzprocesowego COM. Ilustracją poniższej instrukcji są kody źródłowe *approach3* i *approach3a*, a także inne materiały dostępne na stronie sun.iinf.polsl.gliwice.pl/~jfrancik/lectures/com.html.

1. STWORZENIE SPECYFIKACJI *IDL*

Plik IDL zawiera specyfikację tworzonych interfejsów i klas składników (a także biblioteki typów). Wynikiem kompilacji pliku *abc.idl* są między innymi pliki *abc.h* (deklaracje interfejsów w języku C/C++) oraz *abc_i.c* (definicje identyfikatorów *iid* i *clsid*).

[Krok ten eliminuje potrzebę tworzenia pliku nagłówkowego *interface.h*
oraz definiowania explicite identyfikatorów iid / clsid – por. listing approach 3]

2. STWORZENIE APLIKACJI KLIENCKIEJ

Aplikacja kliencka powinna obejmować obydwa pliki wynikowe kompilacji specyfikacji IDL wspomniane w poprzednim punkcie. Aplikacja inicjalizuje system *COM* (*CoInitialize*), zakłada nowy egzemplarz składnika za pomocą funkcji *CoCreateInstance* i otrzymuje wskaźnik na podany interfejs. Wskaźnik na dowolny, inny interfejs aplikacja może otrzymać wywołując funkcję *QueryInterface* już posiadanego interfejsu. Aplikacja jest zobowiązana wywoływać funkcje *AddRef/Release* – zgodnie z obowiązującymi zasadami (por. prezentacja *approach2.pdf*).

3. STWORZENIE MODUŁU DLL SERWERA

Na stworzenie modułu serwera składają się następujące czynności:

1. Wprowadzenie do projektu plików wynikowych kompilacji specyfikacji IDL wyszczególnionych w p. 1.
2. Założenie globalnych liczników: aktywnych składników oraz blokad serwera.
3. Stworzenie klasy składnika. Wszystkie implementowane interfejsy powinny być wyszczególnione jako klasy bazowe. Należy zaimplementować wszystkie funkcje składowe tych interfejsów, a także funkcje składowe interfejsu *IUnknown*: *QueryInterface, AddRef* oraz *Release*.

   [Użycie specyfikacji IDL wymusza, by wszystkie funkcje składowe zwracały HRESULT.
   Listing Approach3 nie spełnia tego wymogu; stosowną poprawkę można znaleźć w listingu Approach3a]
4. Stworzenie klasy implementującej interfejs *IClassFactory* (fabrykę klas). Należy zaimplementować zarówno funkcje składowe interfejsu *IClassFactory* (*CreateInstance* i *LockServer*), jak i wyszczególnione wcześniej funkcje składowe interfejsu *IUnknown*.
5. Implementacja punktu wejściowego DLL (*DllMain*). W obrębie tej funkcji należy zachować uchwyt modułu *DLL HMODULE*.
6. Implementacja obowiązkowych funkcji eksportowanych:
   - *DllCanUnloadNow* – na podstawie wspomnianych w p. 3-2 globalnych liczników ustala, czy serwer może być usunięty z pamięci.
   - *DllGetClassObject* – sprawdza podany *clsid* i tworzy odpowiadającą mu fabrykę klas. Zwraca wskaźnik do wskazanego w parametrze interfejsu fabryki klas.
   - *DllRegisterServer* – na podstawie zachowanego w funkcji *DllMain* uchwytu modułu *DLL* oraz dodatkowych, pomocniczych zmiennych globalnych, przeprowadza rejestrację serwera. Pomocne są tu pliki *registry.h* i *registry.cpp* (© Dale Rogerson) zawierające pomocne funkcje.
   - *DllUnregisterServer* – na podstawie dodatkowych, pomocniczych zmiennych globalnych, przeprowadza wyrejestrowanie serwera. Pomocne są tu pliki *registry.h* i *registry.cpp* (© Dale Rogerson) zawierające pomocne funkcje.
7. Rejestracja modułu w systemie (za pomocą opcji *Tools|Register Control* w *Visual Studio* lub wywołując program regsvr32).

opracowanie: Jarosław Francik