|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| logowydzialu | | Instytut Informatyki Politechniki Śląskiej  Zespół Mikroinformatyki i Teorii Automatów Cyfrowych  **Laboratorium JA** | | logoii | |
| **Rok akademicki** | | **Rodzaj studiów\*: SSI/NSI/NSM** | **Numer ćwiczenia:** | **Grupa** | **Sekcja** |
| **2017/2018** | | **SSI** | **4** | **6** | **1** |
| **Data i godzina planowana ćwiczenia:**  dd/mm/rrrr - gg:mm | | **20/03/2018-11:45** | **Prowadzący**:  OA/AO | **OA** | |
| **Data i godzina wykonania ćwiczenia:**  dd/mm/rrrr - gg:mm | | **20/03/2018-11:45** |
| ***Sprawozdanie*** | | | | | |
| **Temat ćwiczenia:**  Zapoznanie się z programowaniem w j. asemblera MASM 32 w środowisku Windows. | | | | | |
| **Skład sekcji:** | 1.Bartłomiej Krasoń | | | | |

# Cel

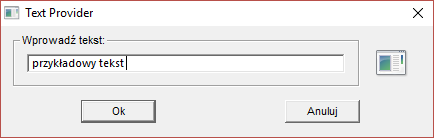
Celem ćwiczenia jest poznanie innych niż Microsoft VC asemblerów i programów debugerów procesorów x86 dla środowiska Windows. Za pomocą narzędzia QEDITOR napisać własny program asemblerowy wykorzystujący podstawowe elementy GUI.

*~Źródło – Instrukcja ćwiczenia LAB4*

# Rozwiązanie

Wykorzystując narzędzie jakim jest QEDITOR dostarczony wraz z pakietem masm32 na mój komputer oraz z pomocą przykładowych programów realizujących podstawowy interfejs GUI w aplikacji, przy użyciu języka asemblera utworzyłem własny, dość podstawowy program „TextProvider”, który pobiera od użytkownika tekst za pomocą kontrolki **Edit** następnie po wciśnięciu ENTER lub kontrolki **Button** „Ok” wyświetla wprowadzony tekst w nowym oknie dialogowym **MessageBox** i kończy pracę programu. Kontrolka **Button** „Anuluj” powoduje zamknięcie aplikacji, po uprzednim powiadomieniu użytkownika. Poniżej prezentuję wynik mojej pracy w postaci:

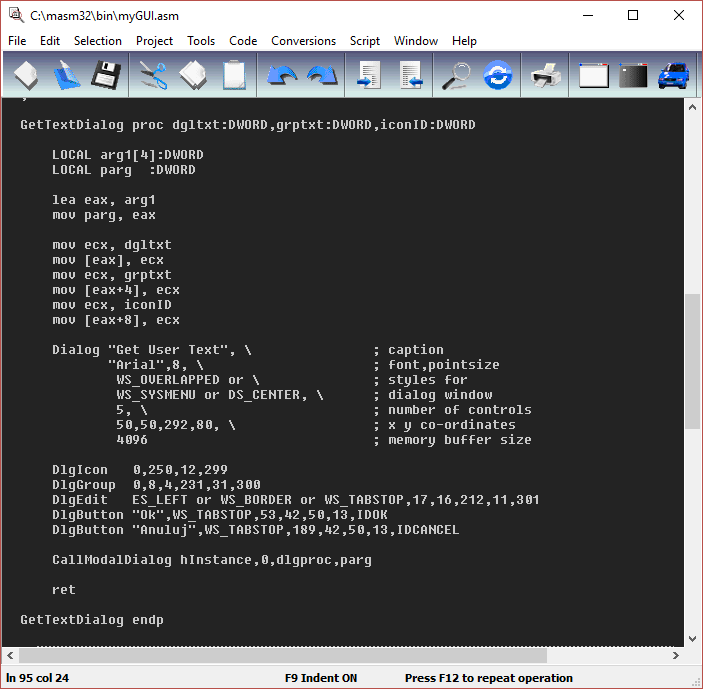
a.) Głównego okna programu:

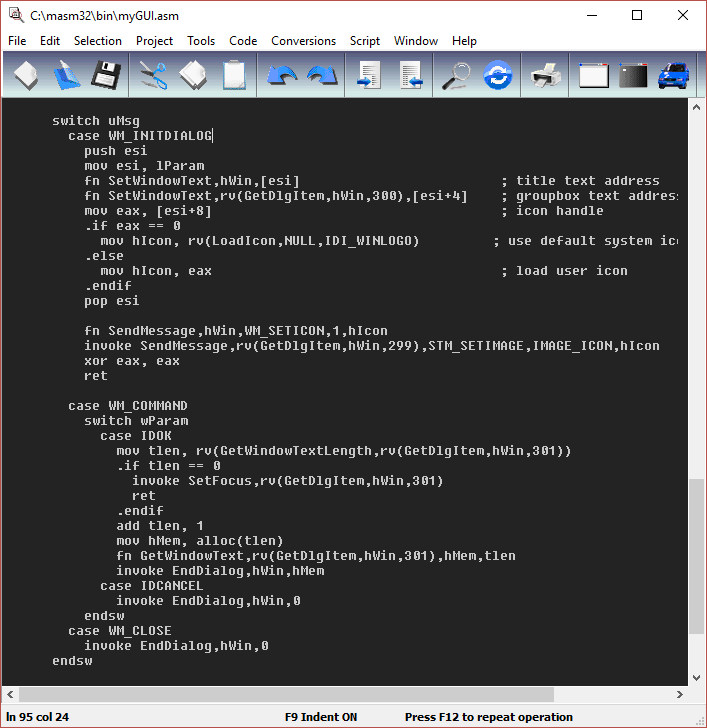


b.) Okna po wprowadzeniu i zatwierdzeniu tekstu:

# messageWindow.PNG

c.) Ciekawszych fragmentów kodu programu:

- funkcja inicjalizującą główne okno programu:

- główna instrukcja switch/case obsługująca „eventy” programu:

# Wnioski

Wykonanie tego ćwiczenia uświadomiło mi, że przy wykorzystaniu odpowiedniej biblioteki, jak i użyciu odpowiednich makr, w języku asemblera można utworzyć aplikację z interfejsem graficznym. Jednakże jest to nieco czasochłonne przedsięwzięcie ze względu na niskopoziomowy charakter asemblera. Do tworzenia małych, nieskomplikowanych pod względem graficznego interfejsu użytkownika aplikacji, rozwiązanie to może się okazać bardzo trafne, gdyż zyskujemy tutaj na optymalizacji czasowej jak i pamięciowej w porównaniu do wyżej poziomowych języków programowania. Natomiast więcej czasu, jak i wysiłku musimy poświęcić na utworzenie takiej aplikacji.