

**Ostatni wykład Eugeniusza Wróbla (7)**  
**Wprowadzenie do drugiego wydania (9)**  
**Rozdział 1. Wprowadzenie (11)**

- 1.1. Co to jest asembler? (11)
- 1.2. Dlaczego programować w języku asemblera? (14)
- 1.3. Dlaczego warto poznać język asemblera? (16)
- 1.4. Wymagane umiejętności (16)
- 1.5. Konwencje stosowane w książce (17)

**Rozdział 2. Pierwszy program w asemblerze (21)**

- 2.1. "Hello, world!" pod kontrolą systemu operacyjnego MS DOS (22)
- 2.2. "Hello, world!" pod kontrolą systemu operacyjnego Windows (25)

**Rozdział 3. Architektura procesorów rodziny x86-32 widziana oczami programisty (33)**

- 3.1. Rejestry procesora 8086 (34)
- 3.2. Zwiększamy rozmiar rejestrów - od procesora 80386 do Intel Core i7 (38)
- 3.3. Zwiększamy liczbę rejestrów - od procesora i486 do Intel Core i7 (39)
- 3.4. Segmentowa organizacja pamięci (44)
- 3.5. Adresowanie argumentów (48)
- 3.6. Adresowanie argumentów w pamięci operacyjnej (49)
- 3.7. Architektura x86-32e (52)

**Rozdział 4. Narzędzia (55)**

- 4.1. Asembler MASM (56)
- 4.2. Program konsolidujący - linker (60)
- 4.3. Programy uruchomieniowe (62)
  - Microsoft CodeView (64)
  - Microsoft WinDbg (67)
  - OllyDbg (68)
- 4.4. Środowiska zintegrowane (70)
  - Microsoft Programmer's WorkBench (PWB) (70)
  - Środowisko zintegrowane MASM32 SDK (71)
  - Środowisko zintegrowane RadASM (74)
  - WinAsm Studio (74)
- 4.5. Microsoft Visual Studio (75)

**Rozdział 5. Lista instrukcji procesorów x86-32 (81)**

- 5.1. Instrukcje ogólne - jednostki stałoprzecinkowej (84)
- 5.2. Koprocesor arytmetyczny - instrukcje jednostki zmiennoprzecinkowej (87)
- 5.3. Instrukcje rozszerzenia MMX (90)
- 5.4. Instrukcje rozszerzenia SSE (93)
- 5.5. Instrukcje rozszerzenia SSE2 (97)
- 5.6. Instrukcje rozszerzenia SSE3, SSSE3 oraz SSE4 (100)
- 5.7. Instrukcje systemowe (101)
- 5.8. Planowane rozszerzenie AVX (102)

## **Rozdział 6. Ogólna struktura programu asemblerowego (105)**

- 6.1. Uproszczone dyrektywy definiujące segmenty (105)
- 6.2. Pełne dyrektywy definiowania segmentów (111)
- 6.3. Dyrektywy pomocnicze (114)

## **Rozdział 7. Definiowanie i stosowanie zmiennych (123)**

- 7.1. Zmienne całkowite (124)
- 7.2. Zmienne zmiennoprzecinkowe (127)
- 7.3. Definiowanie tablic i łańcuchów (128)
- 7.4. Struktury zmiennych (130)
- 7.5. Dyrektywa definiująca pola bitowe (133)

## **Rozdział 8. Podprogramy (137)**

- 8.1. Stos (137)
- 8.2. Wywołanie i organizacja prostych podprogramów (140)
- 8.3. Dyrektywa PROC - ENDP (141)
- 8.4. Parametry wywołania podprogramu (146)
- 8.5. Zmienne lokalne (155)

## **Rozdział 9. Makroinstrukcje oraz dyrektywy asemblacji warunkowej (157)**

- 9.1. Makroinstrukcja definiowana (157)
- 9.2. Dyrektywa LOCAL (162)
- 9.3. Dyrektywy asemblacji warunkowej (163)
- 9.4. Makroinstrukcje niedefiniowane (166)
- 9.5. Makroinstrukcje tekstowe (167)
- 9.6. Makroinstrukcje operujące na łańcuchach (na tekstach) (168)

## **Rozdział 10. Funkcje systemu MS DOS oraz BIOS (171)**

- 10.1. Co ma prawo przerwać wykonanie naszego programu? (171)
- 10.2. Obsługa klawiatury oraz funkcje grafiki na poziomie BIOS (174)
- 10.3. Wywoływanie podprogramów systemu operacyjnego MS DOS (180)

## **Rozdział 11. Programowanie w asemblerze w środowisku Windows (187)**

- 11.1. Systemowe programy biblioteczne (188)
- 11.2. Pierwsze okno (191)
- 11.3. Struktury programowe typu HLL (197)
- 11.4. Program generatora okien Prostart (199)

## **Rozdział 12. Wybrane zagadnienia optymalizacji programu (207)**

- 12.1. Kiedy i co powinniśmy optymalizować w programie? (209)
- 12.2. Optymalizujemy program przygotowany dla procesora x86-32 (211)
  - Modele pamięci - mieszanie kodów 16- i 32-bitowych (211)
  - Wyrównywanie danych (212)

- Pamięć podręczna (213)
  - Unikanie rozgałęzień (skoków) (215)
  - Opóźnienia wynikające z pierwszego wykonania oraz rozwijanie pętli (216)
  - Opóźnienia związane z zapisywaniem i odczytywaniem (217)
- 12.3. Wspieramy proces optymalizacji za pomocą programu Vtune (218)
- 12.4. Na ile różnych sposobów możemy zakodować kopiowanie tablic? (219)
  - Metoda 1.: Z wykorzystaniem instrukcji MOVSB (221)
  - Metoda 2.: Z wykorzystaniem instrukcji MOVSD (221)
  - Metoda 3.: Jawna pętla z instrukcjami MOV (222)
  - Metoda 4.: Pętla z instrukcją MOV, rozwinięta (222)
  - Metoda 5.: Pętla rozwinięta, grupowanie operacji odczytu i zapisu (223)
  - Metoda 6.: Wykorzystujemy rejestry MMX (223)
  - Metoda 7.: Modyfikujemy metodę 6., stosując instrukcje MOVNTQ i SFENCE (224)
  - Metoda 8.: Na początku pętli z poprzedniej metody wprowadzamy instrukcję pobrania wstępnego do pamięci podręcznej (225)
  - Metoda 9.: Wykorzystujemy 128-bitowe rejestry rozszerzenia SSE (225)

### **Rozdział 13. Podział programu na moduły i łączenie modułów zakodowanych w różnych językach programowania (227)**

- 13.1. Jak realizować połączenia międzymodułowe? (228)
- 13.2. Mieszymy moduły przygotowane w różnych językach (232)

### **Rozdział 14. Tworzenie projektu assemblerowego w środowisku Microsoft Visual Studio (239)**

- 14.1. Wstawki assemblerowe w programie uruchamianym w języku C++ (239)
- 14.2. Assemblerowa biblioteka dll w środowisku Microsoft Visual Studio (245)

### **Rozdział 15. Przykładowe programy dla systemu operacyjnego MS DOS (251)**

- 15.1. Pierwsze kroki w prostym trybie graficznym (252)
- 15.2. Pozorujemy głębię (255)
- 15.3. Generowanie fraktali (258)

### **Rozdział 16. Przykładowe programy dla systemu operacyjnego Windows (265)**

- 16.1. Zegarek (265)
- 16.2. Wykorzystanie biblioteki OpenGL (270)
- 16.3. Prosty edytor graficzny (273)

### **Rozdział 17. Biblioteki assemblerowe w środowisku Microsoft Visual Studio (293)**

- 17.1. Tworzenie projektu assemblerowego dla środowiska Visual Studio 2008 (293)
- 17.2. Szyfrowanie (301)
- 17.3. Edytor graficzny (307)
- 17.4. Steganografia (312)

### **Załącznik 1. Interesujące strony w internecie (317)**

### **Załącznik 2. Lista dyrektyw i pseudoinstrukcji języka MASM (319)**

- Z2.1. Dyrektywy określające listę instrukcji procesora (319)
- Z2.2. Organizacja segmentów (321)
- Z2.3. Definiowanie stałych oraz dyrektywy związane z nazwami symbolicznymi (323)
- Z2.4. Definiowanie zmiennych (324)
- Z2.5. Dyrektywy asemblacji warunkowej (326)
- Z2.6. Makroinstrukcje i dyrektywy z nimi związane (327)
- Z2.7. Pseudoinstrukcje typu HLL (329)
- Z2.8. Dyrektywy związane z podprogramami (329)
- Z2.9. Dyrektywy wpływające na kształt listingu asemblacji (330)
- Z2.10. Połączenia międzymodułowe (332)
- Z2.11. Dyrektywy związane z diagnostyką procesu asemblacji (333)
- Z2.12. Inne dyrektywy i pseudoinstrukcje (334)

### **Załącznik 3. Operatory stosowane w języku MASM (337)**

- Z3.1. Operatory stosowane w wyrażeniach obliczanych w czasie asemblacji (337)
- Z3.2. Operatory stosowane w wyrażeniach obliczanych w czasie wykonywania programu (341)

### **Załącznik 4. Symbole predefiniowane (343)**

### **Załącznik 5. Przegląd instrukcji procesora x86-32 (347)**

- Z5.1. Instrukcje ogólne (jednostki stałoprzecinkowej) (347)
- Z5.2. Instrukcje jednostki zmiennoprzecinkowej (koprocessora arytmetycznego) (354)
- Z5.3. Instrukcje rozszerzenia MMX (357)
- Z5.4. Instrukcje rozszerzenia SSE (360)
- Z5.5. Instrukcje rozszerzenia SSE2 (363)
- Z5.6. Instrukcje rozszerzenia SSE3 (367)
- Z5.7. Instrukcje systemowe (368)

### **Załącznik 6. Opis wybranych przerw systemowych BIOS (371)**

- Z6.1. Funkcje obsługi klawiatury wywoływane przerwaniem programowym INT 16h (371)
- Z6.2. Funkcje obsługi karty graficznej wywoływane przerwaniem programowym INT 10h (373)

### **Załącznik 7. Wywołania funkcji systemu operacyjnego MS DOS (379)**

- Z7.1. Funkcje realizujące odczyt lub zapis znaku z układu wejściowego lub wyjściowego (379)
- Z7.2. Funkcje operujące na katalogach (381)
- Z7.3. Operacje na dysku (381)
- Z7.4. Operacje na plikach (zbiorach) dyskowych (383)
- Z7.5. Operacje na rekordach w pliku (385)
- Z7.6. Zarządzanie pamięcią operacyjną (386)
- Z7.7. Funkcje systemowe (387)

- Z7.8. Sterowanie programem (388)
- Z7.9. Funkcje związane z czasem i datą (389)
- Z7.10. Inne funkcje (390)

#### **Załącznik 8. Opis wybranych funkcji API (391)**

- Z8.1. CheckDlgButton (391)
- Z8.2. CloseHandle (392)
- Z8.3. CopyFile (393)
- Z8.4. CreateFile (394)
- Z8.5. CreateWindowEx (396)
- Z8.6. DeleteFile (399)
- Z8.7. ExitProcess (399)
- Z8.8. GetFileSize (400)
- Z8.9. MessageBox (400)
- Z8.10. ShowWindow (403)

#### **Załącznik 9. Tablica kodów ASCII oraz kody klawiszy (405)**

- Z9.1. Kody ASCII (405)
- Z9.2. Kody klawiszy (405)

#### **Załącznik 10. FTP wydawnictwa (411) Skorowidz (413)**