

Rozdział 1. Wprowadzenie (7)

- 1.1. Co to jest assembler? (7)
- 1.2. Dlaczego programować w języku assemblera? (10)
- 1.3. Dlaczego warto poznać język assemblera? (12)
- 1.4. Wymagane umiejętności (12)
- 1.5. Konwencje stosowane w książce (13)

Rozdział 2. Zaczynamy typowo - wiedząc niewiele, uruchamiamy nasz pierwszy program (17)

- 2.1. "Hello, world!" pod systemem operacyjnym MS DOS (18)
- 2.2. "Hello, world!" pod systemem operacyjnym Windows (22)

Rozdział 3. Wracamy do podstaw - poznajemy minimum wiedzy na temat architektury procesorów 80x86 (29)

- 3.1. Rejestry procesora 8086 (30)
- 3.2. Zwiększamy rozmiar rejestrów - od procesora 80386 do Pentium 4 (33)
- 3.3. Zwiększamy liczbę rejestrów - od procesora 80486 do Pentium 4 (35)
- 3.4. Segmentowa organizacja pamięci (39)
- 3.5. Adresowanie argumentów (43)
- 3.6. Adresowanie argumentów w pamięci operacyjnej (44)

Rozdział 4. Poznajemy narzędzia (47)

- 4.1. Assembler MASM (49)
- 4.2. Program konsolidujący - linker (52)
- 4.3. Programy uruchomieniowe (54)
- 4.4. Wszystkie potrzebne narzędzia razem, czyli środowiska zintegrowane (62)

Rozdział 5. Nadmiar możliwości, z którym trudno sobie poradzić - czyli lista instrukcji procesora (67)

- 5.1. Instrukcje ogólne - jednostki stałoprzecinkowej (70)
- 5.2. Koprocesor arytmetyczny - instrukcje jednostki zmiennoprzecinkowej (73)
- 5.3. Instrukcje rozszerzenia MMX (75)
- 5.4. Instrukcje rozszerzenia SSE (78)
- 5.5. Instrukcje rozszerzenia SSE2 (82)
- 5.6. Instrukcje rozszerzenia SSE3 (85)
- 5.7. Instrukcje systemowe (85)

Rozdział 6. Wracamy do ogólnej struktury programu assemblerowego (87)

- 6.1. Uprozczone dyrektywy definiujące segmenty (87)
- 6.2. Pełne dyrektywy definiowania segmentów (92)
- 6.3. Spróbujmy drobną część tej wiedzy zastosować w prostym programie, a przy okazji poznajmy nowe pomocnicze dyrektywy (96)

Rozdział 7. Ważna rzecz w każdym języku programowania - definiowanie i stosowanie zmiennych (105)

- 7.1. Zmienne całkowite (106)
- 7.2. Zmienne zmiennoprzecinkowe (109)
- 7.3. Definiowanie tablic i łańcuchów (110)
- 7.4. Struktury zmiennych (114)
- 7.5. Dyrektywa definiująca pola bitowe (117)

Rozdział 8. Podprogramy (119)

- 8.1. Stos (119)
- 8.2. Wywołanie i organizacja prostych podprogramów (122)
- 8.3. Poznajemy dyrektywę PROC-ENDP (123)
- 8.4. Parametry wywołania podprogramu (128)
- 8.5. Zmienne lokalne (137)

Rozdział 9. Oddalamy się od assemblera w kierunku języków wyższego poziomu, czyli użycie makroinstrukcji oraz dyrektyw asemlacji warunkowej (139)

- 9.1. Makroinstrukcja definiowana (139)
- 9.2. Dyrektywa LOCAL (144)
- 9.3. Dyrektywy asemlacji warunkowej (144)
- 9.4. Makroinstrukcje niedefiniowane (148)
- 9.5. Makroinstrukcje tekstowe (149)
- 9.6. Makroinstrukcje operujące na łańcuchach (na tekstach) (150)

Rozdział 10. Czy obsługę wszystkich urządzeń komputera musimy wykonać sami? Funkcje systemu MS DOS oraz BIOS (153)

- 10.1. Co ma prawo przerwać wykonanie naszego programu? (154)
- 10.2. Obsługa klawiatury oraz funkcje grafiki na poziomie BIOS (156)
- 10.3. Wywoływanie podprogramów systemu operacyjnego MS DOS (163)

Rozdział 11. Obalamy mity programując w assemblerze pod systemem operacyjnym Windows (169)

- 11.1. Systemowe programy biblioteczne (170)
- 11.2. Najprawdziwsze pierwsze okno (173)
- 11.3. Struktury programowe HLL - to też jest assembler! (178)
- 11.4. Idziemy jeden krok dalej i wykorzystujemy program generatora okien Prostart (180)

Rozdział 12. Czy możemy przyspieszyć działanie naszego programu? Wybrane zagadnienia optymalizacji programu (189)

- 12.1. Kiedy i co w programie powinniśmy optymalizować? (191)
- 12.2. Optymalizujemy program przygotowany dla procesora Pentium 4 (193)
- 12.3. Wspieramy proces optymalizacji programem Vtune (200)
- 12.4. Na ile różnych sposobów możemy zakodować kopiowanie tablic? (201)

Rozdział 13. Dzielimy program na moduły i łączymy moduły zakodowane w różnych językach programowania (209)

- 13.1. Jak realizować połączenia międzymodułowe? (210)
- 13.2. Mieszmamy moduły przygotowane w różnych językach (214)

Rozdział 14. Przykładowe programy (MS DOS) (219)

- 14.1. Identyfikujemy procesor (219)
- 14.2. Wchodzimy w świat grafiki - nieco patriotycznie (225)
- 14.3. Program rezydentny, czyli namiastka wielozadaniowości (228)
- 14.4. Pozorujemy głębię (233)
- 14.5. Wyższa graficzna szkoła jazdy ze zmiennymi zespolonymi (236)

Rozdział 15. Przykładowe programy (Windows) (243)

- 15.1. Zegarek (243)
- 15.2. Dotknięcie grafiki przez duże "G" (248)
- 15.3. Przekształcamy mapę bitową (250)

Załącznik 1. Interesujące strony w internecie (271)

Załącznik 2. Lista dyrektyw i pseudoinstrukcji języka MASM (275)

- Z2.1. Dyrektywy określające listę instrukcji procesora (275)
- Z2.2. Organizacja segmentów (277)
- Z2.3. Definiowanie stałych oraz dyrektywy związane z nazwami symbolicznymi (279)
- Z2.4. Definiowanie zmiennych (280)
- Z2.4. Dyrektywy asemblacji warunkowej (282)
- Z2.5. Makroinstrukcje i dyrektywy nimi związane (283)
- Z2.6. Pseudoinstrukcje typu HLL (285)
- Z2.7. Dyrektywy związane z podprogramami (286)
- Z2.8. Dyrektywy wpływające na kształt listingu asemblacji (287)
- Z2.9. Połączenia międzymodułowe (289)
- Z2.10. Dyrektywy związane z diagnostyką procesu asemblacji (290)
- Z2.11. Inne dyrektywy i pseudoinstrukcje (291)

Załącznik 3. Operatory stosowane w języku MASM (293)

- Z3.1. Operatory stosowane w wyrażeniach obliczanych w czasie asemblacji (293)
- Z3.2. Operatory stosowane w wyrażeniach obliczanych w czasie wykonywania programu (297)

Załącznik 4. Symbole predefiniowane (299)

Załącznik 5. Przegląd instrukcji procesora Pentium 4 (303)

- Z5.1. Instrukcje ogólne (jednostki stałoprzecinkowej) (303)
- Z5.2. Instrukcje jednostki zmiennoprzecinkowej (koprocessora arytmetycznego) (309)
- Z5.3. Instrukcje rozszerzenia MMX (313)
- Z5.4. Instrukcje rozszerzenia SSE (315)
- Z5.5. Instrukcje rozszerzenia SSE2 (319)

- Z5.6. Instrukcje rozszerzenia SSE3 (323)
- Z5.7. Instrukcje systemowe (325)

Załącznik 6. Opis wybranych przerw systemowych BIOS (327)

- Z6.1. Funkcje obsługi klawiatury wywoływane przerwaniem programowym INT 16h (327)
- Z6.2. Funkcje obsługi karty graficznej wywoływane przerwaniem programowym INT 10h. (329)

Załącznik 7. Wywołania funkcji systemu operacyjnego MS DOS (335)

- Z7.1. Funkcje realizujące odczyt lub zapis znaku z układu wejściowego lub wyjściowego (335)
- Z7.2. Funkcje operujące na katalogach (337)
- Z7.3. Operacje na dysku (337)
- Z7.4. Operacje na plikach (zbiorach) dyskowych (339)
- Z7.5. Operacje na rekordach w pliku (341)
- Z7.6. Zarządzanie pamięcią operacyjną (342)
- Z7.7. Funkcje systemowe (342)
- Z7.8. Sterowanie programem (344)
- Z7.9. Funkcje związane z czasem i datą (345)
- Z7.10 Inne funkcje (345)

Załącznik 8. Opis wybranych funkcji API (347)

- Z8.1. CheckDlgButton (347)
- Z8.2. CloseHandle (348)
- Z8.3 CopyFile (349)
- Z8.4. CreateFile (350)
- Z8.5. CreateWindowEx (352)
- Z8.6. DeleteFile (355)
- Z8.7. ExitProcess (355)
- Z8.8. GetFileSize (356)
- Z8.9. MessageBox (357)
- Z8.10. ShowWindow (359)

Załącznik 9. Tablica kodów ASCII oraz kody klawiszy (361)

- Z9.1. Kody ASCII (361)
- Z9.2. Kody klawiszy (361)

Załącznik 10. Program Segment Prefix (PSP) (367)

Załącznik 11. Płyta CD załączona do książki (369)

Skorowidz (371)