

Rozdział 1. Wstęp do języka assemblerowego (13)

- 1.1. Wprowadzenie (13)
- 1.2. Anatomia programu HLA (13)
- 1.3. Uruchamianie pierwszego programu HLA (16)
- 1.4. Podstawowe deklaracje danych programu HLA (17)
- 1.5. Wartości logiczne (19)
- 1.6. Wartości znakowe (20)
- 1.7. Rodzina procesorów 80x86 firmy Intel (21)
 - o 1.7.1. Podsystem obsługi pamięci (24)
- 1.8. Podstawowe instrukcje maszynowe (26)
- 1.9. Podstawowe struktury sterujące wykonaniem programu HLA (30)
 - o 1.9.1. Wyrażenia logiczne w instrukcjach HLA (31)
 - o 1.9.2. Instrukcje if..then..elseif..else..endif języka HLA (33)
 - o 1.9.3. Iloczyn, suma i negacja w wyrażeniach logicznych (35)
 - o 1.9.4. Instrukcja while (37)
 - o 1.9.5. Instrukcja for (38)
 - o 1.9.6. Instrukcja repeat (39)
 - o 1.9.7. Instrukcje break oraz breakif (40)
 - o 1.9.8. Instrukcja forever (40)
 - o 1.9.9. Instrukcje try, exception oraz endtry (41)
- 1.10. Biblioteka standardowa języka HLA - wprowadzenie (44)
 - o 1.10.1. Stałe predefiniowane w module stdio (46)
 - o 1.10.2. Standardowe wejście i wyjście programu (46)
 - o 1.10.3. Procedura stdout.newln (47)
 - o 1.10.4. Procedury stdout.putiN (47)
 - o 1.10.5. Procedury stdout.putiNSize (48)
 - o 1.10.6. Procedura stdout.put (49)
 - o 1.10.7. Procedura stdin.getc (51)
 - o 1.10.8. Procedury stdin.getiN (52)
 - o 1.10.9. Procedury stdin.readLn i stdin.flushInput (53)
 - o 1.10.10. Procedura stdin.get (54)
- 1.11. Jeszcze o ochronie wykonania kodu w bloku try..endtry (55)
 - o 1.11.1. Zagnieżdżone bloki try..endtry (56)
 - o 1.11.2. Klauzula unprotected bloku try..endtry (58)
 - o 1.11.3. Klauzula anyexception bloku try..endtry (61)
 - o 1.11.4. Instrukcja try..endtry i rejestry (61)
- 1.12. Język assemblerowy a język HLA (63)
- 1.13. Źródła informacji dodatkowych (64)

Rozdział 2. Reprezentacja danych (65)

- 2.1. Wprowadzenie (65)
- 2.2. Systemy liczbowe (66)
 - o 2.2.1. System dziesiętny - przypomnienie (66)
 - o 2.2.2. System dwójkowy (66)
 - o 2.2.3. Formaty liczb dwójkowych (68)
- 2.3. System szesnastkowy (69)
- 2.4. Organizacja danych (72)
 - o 2.4.1. Bity (72)

- o 2.4.2. Półbajty (73)
- o 2.4.3. Bajty (73)
- o 2.4.4. Słowa (75)
- o 2.4.5. Podwójne słowa (76)
- o 2.4.6. Słowa poczwórne i długie (77)
- 2.5. Operacje arytmetyczne na liczbach dwójkowych i szesnastkowych (77)
- 2.6. Jeszcze o liczbach i ich reprezentacji (78)
- 2.7. Operacje logiczne na bitach (81)
- 2.8. Operacje logiczne na liczbach dwójkowych i ciągach bitów (84)
- 2.9. Liczby ze znakiem i bez znaku (86)
- 2.10. Rozszerzanie znakiem, rozszerzanie zerem, skracanie, przycinanie (91)
- 2.11. Przesunięcia i obroty (95)
- 2.12. Pola bitowe i dane spakowane (99)
- 2.13. Wprowadzenie do arytmetyki zmiennoprzecinkowej (104)
 - o 2.13.1. Formaty zmiennoprzecinkowe przyjęte przez IEEE (108)
 - o 2.13.2. Obsługa liczb zmiennoprzecinkowych w języku HLA (112)
- 2.14. Reprezentacja liczb BCD (115)
- 2.15. Znaki (117)
 - o 2.15.1. Zestaw znaków ASCII (117)
 - o 2.15.2. Obsługa znaków ASCII w języku HLA (121)
- 2.16. Zestaw znaków Unicode (125)
- 2.17. Źródła informacji dodatkowych (126)

Rozdział 3. Dostęp do pamięci i jej organizacja (127)

- 3.1. Wprowadzenie (127)
- 3.2. Tryby adresowania procesorów 80x86 (127)
 - o 3.2.1. Adresowanie przez rejestr (128)
 - o 3.2.2. 32-bitowe tryby adresowania procesora 80x86 (129)
- 3.3. Organizacja pamięci fazy wykonania (135)
 - o 3.3.1. Obszar kodu (137)
 - o 3.3.2. Obszar zmiennych statycznych (139)
 - o 3.3.3. Obszar niemodyfikowalny (140)
 - o 3.3.4. Obszar danych niezainicjalizowanych (141)
 - o 3.3.5. Atrybut @nostorage (141)
 - o 3.3.6. Sekcja deklaracji var (142)
 - o 3.3.7. Rozmieszczenie sekcji deklaracji danych w programie HLA (143)
- 3.4. Przydział pamięci dla zmiennych w programach HLA (144)
- 3.5. Wyrównanie danych w programach HLA (146)
- 3.6. Wyrażenia adresowe (149)
- 3.7. Koercja typów (151)
- 3.8. Koercja typu rejestru (154)
- 3.9. Pamięć obszaru stosu oraz instrukcje push i pop (155)
 - o 3.9.1. Podstawowa postać instrukcji push (155)
 - o 3.9.2. Podstawowa postać instrukcji pop (157)
 - o 3.9.3. Zachowywanie wartości rejestrów za pomocą instrukcji push i pop (158)
 - o 3.9.4. Stos jako kolejka LIFO (159)
 - o 3.9.5. Pozostałe wersje instrukcji obsługi stosu (161)
 - o 3.9.6. Usuwanie danych ze stosu bez ich zdejmowania (163)
 - o 3.9.7. Odwoływanie się do danych na stosie bez ich zdejmowania (165)

- 3.10. Dynamiczny przydział pamięci - obszar pamięci sterty (166)
- 3.11. Instrukcje inc oraz dec (171)
- 3.12. Pobieranie adresu obiektu (171)
- 3.13. Źródła informacji dodatkowych (172)

Rozdział 4. Stałe, zmienne i typy danych (173)

- 4.1. Wprowadzenie (173)
- 4.2. Kilka dodatkowych instrukcji: intmul, bound i into (174)
- 4.3. Typ tbyte (178)
- 4.4. Deklaracje stałych i zmiennych w języku HLA (178)
 - o 4.4.1. Typy stałych (182)
 - o 4.4.2. Literały stałych łańcuchowych i znakowych (183)
 - o 4.4.3. Stałe łańcuchowe i napisowe w sekcji const (185)
 - o 4.4.4. Wyrażenia stałowartościowe (186)
 - o 4.4.5. Wielokrotne sekcje const i ich kolejność w programach HLA (189)
 - o 4.4.6. Sekcja val programu HLA (190)
 - o 4.4.7. Modyfikowanie obiektów sekcji val w wybranym miejscu kodu źródłowego programu (191)
- 4.5. Sekcja type programu HLA (192)
- 4.6. Typy wyliczeniowe w języku HLA (193)
- 4.7. Typy wskaźnikowe (194)
 - o 4.7.1. Wskaźniki w języku assemblerowym (196)
 - o 4.7.2. Deklarowanie wskaźników w programach HLA (197)
 - o 4.7.3. Stałe wskaźnikowe i wyrażenia stałych wskaźnikowych (197)
 - o 4.7.4. Zmienne wskaźnikowe a dynamiczny przydział pamięci (199)
 - o 4.7.5. Typowe błędy stosowania wskaźników (200)
- 4.8. Moduł chars.hhf biblioteki standardowej HLA (205)
- 4.9. Złożone typy danych (207)
- 4.10. Łańcuchy znaków (208)
- 4.11. Łańcuchy w języku HLA (210)
- 4.12. Odwołania do poszczególnych znaków łańcucha (217)
- 4.13. Moduł strings biblioteki standardowej HLA i procedury manipulacji łańcuchami (219)
- 4.14. Konwersje wewnątrzpamięciowe (231)
- 4.15. Zbiory znaków (232)
- 4.16. Implementacja zbiorów znaków w języku HLA (233)
- 4.17. Literały, stałe i wyrażenia zbiorów znaków w języku HLA (235)
- 4.18. Operator in w wyrażeniach logicznych wysokopoziomowego rozszerzenia języka HLA (237)
- 4.19. Obsługa zbiorów znaków w bibliotece standardowej HLA (237)
- 4.20. Wykorzystywanie zbiorów znaków w programach HLA (241)
- 4.21. Tablice (243)
- 4.22. Deklarowanie tablic w programach HLA (244)
- 4.23. Literały tablicowe (245)
- 4.24. Odwołania do elementów tablicy jednowymiarowej (246)
 - o 4.24.1. Porządkowanie tablicy wartości (248)
- 4.25. Tablice wielowymiarowe (250)
 - o 4.25.1. Wierszowy układ elementów tablicy (251)
 - o 4.25.2. Kolumnowy układ elementów tablicy (255)

- 4.26. Przydział pamięci dla tablic wielowymiarowych (256)
- 4.27. Odwołania do elementów tablic wielowymiarowych w języku assemblerowym (258)
- 4.28. Duże tablice i MASM (tylko dla programistów systemu Windows) (259)
- 4.29. Rekordy (struktury) (260)
- 4.30. Stałe rekordowe (263)
- 4.31. Tablice rekordów (264)
- 4.32. Wykorzystanie tablic i rekordów w roli pól rekordów (265)
- 4.33. Ingerowanie w przesunięcia pól rekordów (269)
- 4.34. Wyrównanie pól w ramach rekordu (270)
- 4.35. Wskaźniki na rekordy (271)
- 4.36. Unie (273)
- 4.37. Unie anonimowe (275)
- 4.38. Typy wariantowe (276)
- 4.39. Stałe unii (277)
- 4.40. Przestrzenie nazw (278)
- 4.41. Tablice dynamiczne w języku assemblerowym (281)
- 4.42. Obsługa tablic w bibliotece standardowej języka HLA (284)
- 4.43. Źródła informacji dodatkowych (287)

Rozdział 5. Procedury i moduły (289)

- 5.1. Wprowadzenie (289)
- 5.2. Procedury (289)
- 5.3. Zachowywanie stanu systemu (292)
- 5.4. Przedwczesny powrót z procedury (296)
- 5.5. Zmienne lokalne (297)
- 5.6. Symbole lokalne i globalne obiektów innych niż zmienne (303)
- 5.7. Parametry (304)
 - o 5.7.1. Przekazywanie przez wartość (305)
 - o 5.7.2. Przekazywanie przez adres (308)
- 5.8. Funkcje i wartości funkcji (311)
 - o 5.8.1. Zwracanie wartości funkcji (312)
 - o 5.8.2. Złożenie instrukcji języka HLA (313)
 - o 5.8.3. Atrybut @returns procedur języka HLA (316)
- 5.9. Rekurencja (318)
- 5.10. Deklaracje zapowiadające (322)
- 5.11. Procedury w ujęciu niskopoziomowym - instrukcja call (323)
- 5.12. Rola stosu w procedurach (325)
- 5.13. Rekordy aktywacji (328)
- 5.14. Standardowa sekwencja wejścia do procedury (331)
- 5.15. Standardowa sekwencja wyjścia z procedury (333)
- 5.16. Niskopoziomowa implementacja zmiennych automatycznych (334)
- 5.17. Niskopoziomowa implementacja parametrów procedury (336)
 - o 5.17.1. Przekazywanie argumentów w rejestrach (337)
 - o 5.17.2. Przekazywanie argumentów w kodzie programu (340)
 - o 5.17.3. Przekazywanie argumentów przez stos (342)
- 5.18. Wskaźniki na procedury (365)
- 5.19. Parametry typu procedurowego (368)
- 5.20. Nietypowane parametry wskaźnikowe (370)

- 5.21. Zarządzanie dużymi projektami programistycznymi (371)
- 5.22. Dyrektywa #include (372)
- 5.23. Unikanie wielokrotnego włączania do kodu tego samego pliku (374)
- 5.24. Moduły a atrybut @external (375)
 - o 5.24.1. Działanie atrybutu @external (380)
 - o 5.24.2. Pliki nagłówkowe w programach HLA (382)
- 5.25. Jeszcze o problemie zaśmiecania przestrzeni nazw (384)
- 5.26. Źródła informacji dodatkowych (386)

Rozdział 6. Arytmetyka (389)

- 6.1. Wprowadzenie (389)
- 6.2. Zestaw instrukcji arytmetycznych procesora 80x86 (389)
 - o 6.2.1. Instrukcje mul i imul (389)
 - o 6.2.2. Instrukcje div i idiv (393)
 - o 6.2.3. Instrukcja cmp (396)
 - o 6.2.4. Instrukcje setXX (401)
 - o 6.2.5. Instrukcja test (403)
- 6.3. Wyrażenia arytmetyczne (404)
 - o 6.3.1. Proste przypisania (405)
 - o 6.3.2. Proste wyrażenia (406)
 - o 6.3.3. Wyrażenia złożone (408)
 - o 6.3.4. Operatory przemienne (413)
- 6.4. Wyrażenia logiczne (414)
- 6.5. Idiomy maszynowe a idiomy arytmetyczne (417)
 - o 6.5.1. Mnożenie bez stosowania instrukcji mul, imul i intmul (417)
 - o 6.5.2. Dzielenie bez stosowania instrukcji div i idiv (419)
 - o 6.5.3. Zliczanie modulo n za pośrednictwem instrukcji and (420)
 - o 6.5.4. Nieostrożne korzystanie z idiomów maszynowych (420)
- 6.6. Arytmetyka zmiennoprzecinkowa (421)
 - o 6.6.1. Rejestry jednostki zmiennoprzecinkowej (421)
 - o 6.6.2. Typy danych jednostki zmiennoprzecinkowej (429)
 - o 6.6.3. Zestaw instrukcji jednostki zmiennoprzecinkowej (430)
 - o 6.6.4. Instrukcje przemieszczania danych (431)
 - o 6.6.5. Instrukcje konwersji (433)
 - o 6.6.6. Instrukcje arytmetyczne (436)
 - o 6.6.7. Instrukcje porównań (442)
 - o 6.6.8. Instrukcje ładowania stałych na stos koprocesora (445)
 - o 6.6.9. Instrukcje funkcji przestępnych (445)
 - o 6.6.10. Pozostałe instrukcje jednostki zmiennoprzecinkowej (447)
 - o 6.6.11. Instrukcje operacji całkowitoliczbowych (449)
- 6.7. Tłumaczenie wyrażeń arytmetycznych na kod maszynowy jednostki zmiennoprzecinkowej (449)
 - o 6.7.1. Konwersja notacji wrostkowej do odwrotnej notacji polskiej (451)
 - o 6.7.2. Konwersja odwrotnej notacji polskiej do kodu języka assemblerowego (453)
- 6.8. Obsługa arytmetyki zmiennoprzecinkowej w bibliotece standardowej języka HLA (455)
 - o 6.8.1. Funkcje stdin.getf i fileio.getf (455)
 - o 6.8.2. Funkcje trygonometryczne modułu math (455)

- o 6.8.3. Funkcje wykładnicze i logarytmiczne modułu math (456)
- 6.9. Podsumowanie (458)

Rozdział 7. Niskopoziomowe struktury sterujące wykonaniem programu (459)

- 7.1. Wprowadzenie (459)
- 7.2. Struktury sterujące niskiego poziomu (460)
- 7.3. Etykiety instrukcji (460)
- 7.4. Bezwarunkowy skok do instrukcji (instrukcja jmp) (462)
- 7.5. Instrukcje skoku warunkowego (465)
- 7.6. Struktury sterujące "średniego" poziomu - jt i jf (468)
- 7.7. Implementacja popularnych struktur sterujących w języku assemblerowym (469)
- 7.8. Wstęp do podejmowania decyzji (469)
 - o 7.8.1. Instrukcje if..then..else (471)
 - o 7.8.2. Tłumaczenie instrukcji if języka HLA na język assemblerowy (475)
 - o 7.8.3. Obliczanie wartości złożonych wyrażeń logicznych - metoda pełnego szacowania wartości wyrażenia (480)
 - o 7.8.4. Skrócone szacowanie wyrażeń logicznych (481)
 - o 7.8.5. Wady i zalety metod szacowania wartości wyrażeń logicznych (483)
 - o 7.8.6. Efektywna implementacja instrukcji if w języku assemblerowym (485)
 - o 7.8.7. Instrukcje wyboru (490)
- 7.9. Skoki pośrednie a automaty stanów (500)
- 7.10. Kod spaghetti (503)
- 7.11. Pętle (504)
 - o 7.11.1. Pętle while (505)
 - o 7.11.2. Pętle repeat..until (506)
 - o 7.11.3. Pętle nieskończone (508)
 - o 7.11.4. Pętle for (508)
 - o 7.11.5. Instrukcje break i continue (509)
 - o 7.11.6. Pętle a rejestry (513)
- 7.12. Optymalizacja kodu (514)
 - o 7.12.1. Obliczanie warunku zakończenia pętli na końcu pętli (515)
 - o 7.12.2. Zliczanie licznika pętli wstecz (517)
 - o 7.12.3. Wstępne obliczanie niezmienników pętli (518)
 - o 7.12.4. Rozciąganie pętli (519)
 - o 7.12.5. Zmienne indukcyjne (521)
- 7.13. Mieszane struktury sterujące w języku HLA (522)
- 7.14. Źródła informacji dodatkowych (524)

Rozdział 8. Pliki (525)

- 8.1. Wprowadzenie (525)
- 8.2. Organizacja plików (525)
 - o 8.2.1. Pliki jako listy rekordów (526)
 - o 8.2.2. Pliki tekstowe a pliki binarne (528)
- 8.3. Pliki sekwencyjne (530)
- 8.4. Pliki dostępu swobodnego (538)
- 8.5. Indeksowany sekwencyjny dostęp do pliku (ISAM) (543)
- 8.6. Przycinanie pliku (546)
- 8.7. Źródła informacji dodatkowych (548)

Rozdział 9. Zaawansowane obliczenia w języku asemblerowym (549)

- 9.1. Wprowadzenie (549)
- 9.2. Operacje o wielokrotnionej precyzji (550)
 - o 9.2.1. Obsługa operacji wielokrotnionej precyzji w bibliotece standardowej języka HLA (550)
 - o 9.2.2. Dodawanie liczb wielokrotnionej precyzji (553)
 - o 9.2.3. Odejmowanie liczb wielokrotnionej precyzji (556)
 - o 9.2.4. Porównanie wartości o wielokrotnionej precyzji (558)
 - o 9.2.5. Mnożenie operandów wielokrotnionej precyzji (562)
 - o 9.2.6. Dzielenie wartości wielokrotnionej precyzji (565)
 - o 9.2.7. Negacja operandów wielokrotnionej precyzji (575)
 - o 9.2.8. Iloczyn logiczny operandów wielokrotnionej precyzji (577)
 - o 9.2.9. Suma logiczna operandów wielokrotnionej precyzji (577)
 - o 9.2.10. Suma wyłączająca operandów wielokrotnionej precyzji (578)
 - o 9.2.11. Inwersja operandów wielokrotnionej precyzji (578)
 - o 9.2.12. Przesunięcia bitowe operandów wielokrotnionej precyzji (578)
 - o 9.2.13. Obroty operandów wielokrotnionej precyzji (583)
 - o 9.2.14. Operandy wielokrotnionej precyzji w operacjach wejścia-wyjścia (583)
- 9.3. Manipulowanie operandami różnych rozmiarów (604)
- 9.4. Arytmetyka liczb dziesiętnych (606)
 - o 9.4.1. Literały liczb BCD (608)
 - o 9.4.2. Instrukcje maszynowe daa i das (608)
 - o 9.4.3. Instrukcje maszynowe aaa, aas, aam i aad (610)
 - o 9.4.4. Koprocessor a arytmetyka spakowanych liczb dziesiętnych (612)
- 9.5. Obliczenia w tabelach (615)
 - o 9.5.1. Wyszukiwanie w tabeli wartości funkcji (615)
 - o 9.5.2. Dopasowywanie dziedziny (620)
 - o 9.5.3. Generowanie tabel wartości funkcji (621)
 - o 9.5.4. Wydajność odwołań do tabel przeglądowych (625)
- 9.6. Źródła informacji dodatkowych (625)

Rozdział 10. Makrodefinicje i język czasu kompilacji (627)

- 10.1. Wprowadzenie (627)
- 10.2. Język czasu kompilacji - wstęp (627)
- 10.3. Instrukcje #print i #error (629)
- 10.4. Stałe i zmienne czasu kompilacji (631)
- 10.5. Wyrażenia i operatory czasu kompilacji (632)
- 10.6. Funkcje czasu kompilacji (635)
 - o 10.6.1. Funkcje czasu kompilacji - konwersja typów (636)
 - o 10.6.2. Funkcje czasu kompilacji - obliczenia numeryczne (638)
 - o 10.6.3. Funkcje czasu kompilacji - klasyfikacja znaków (638)
 - o 10.6.4. Funkcje czasu kompilacji - manipulacje łańcuchami znaków (639)
 - o 10.6.5. Funkcje czasu kompilacji - dopasowywanie wzorców (639)
 - o 10.6.6. Odwołania do tablicy symboli (641)
 - o 10.6.7. Pozostałe funkcje czasu kompilacji (643)
 - o 10.6.8. Konwersja typu stałych napisowych (643)
- 10.7. Kompilacja warunkowa (645)

- 10.8. Kompilacja wielokrotna (pętla czasu kompilacji) (650)
- 10.9. Makrodefinicje (procedury czasu kompilacji) (653)
 - o 10.9.1. Makrodefinicje standardowe (654)
 - o 10.9.2. Argumenty makrodefinicji (656)
 - o 10.9.3. Symbole lokalne makrodefinicji (663)
 - o 10.9.4. Makrodefinicje jako procedury czasu kompilacji (666)
 - o 10.9.5. Symulowane przeciążanie funkcji (667)
- 10.10. Tworzenie programów czasu kompilacji (672)
 - o 10.10.1. Generowanie tabel wartości funkcji (673)
 - o 10.10.2. Rozciąganie pętli (677)
- 10.11. Stosowanie makrodefinicji w osobnych plikach kodu źródłowego (679)
- 10.12. Źródła informacji dodatkowych (679)

Rozdział 11. Manipulowanie bitami (681)

- 11.1. Wprowadzenie (681)
- 11.2. Czym są dane bitowe? (681)
- 11.3. Instrukcje manipulujące bitami (683)
- 11.4. Znacznik przeniesienia w roli akumulatora bitów (692)
- 11.5. Wstawianie i wyodrębnianie łańcuchów bitów (693)
- 11.6. Scalanie zbiorów bitów i rozpraszanie łańcuchów bitowych (696)
- 11.7. Spakowane tablice łańcuchów bitowych (699)
- 11.8. Wyszukiwanie bitów (701)
- 11.9. Zliczanie bitów (704)
- 11.10. Odwracanie łańcucha bitów (707)
- 11.11. Scalanie łańcuchów bitowych (709)
- 11.12. Wyodrębnianie łańcuchów bitów (710)
- 11.13. Wyszukiwanie wzorca bitowego (712)
- 11.14. Moduł bits biblioteki standardowej HLA (713)
- 11.15. Źródła informacji dodatkowych (715)

Rozdział 12. Operacje łańcuchowe (717)

- 12.1. Wprowadzenie (717)
- 12.2. Instrukcje łańcuchowe procesorów 80x86 (717)
 - o 12.2.1. Sposób działania instrukcji łańcuchowych (718)
 - o 12.2.2. Przedrostki instrukcji łańcuchowych - repX (719)
 - o 12.2.3. Znacznik kierunku (719)
 - o 12.2.4. Instrukcja movs (721)
 - o 12.2.5. Instrukcja cmps (727)
 - o 12.2.6. Instrukcja scas (731)
 - o 12.2.7. Instrukcja stos (732)
 - o 12.2.8. Instrukcja lods (733)
 - o 12.2.9. Instrukcje lods i stos w złożonych operacjach łańcuchowych (733)
- 12.3. Wydajność instrukcji łańcuchowych procesorów 80x86 (734)
- 12.4. Źródła informacji dodatkowych (735)

Rozdział 13. Instrukcje MMX (737)

- 13.1. Wprowadzenie (737)

- 13.2. Sprawdzanie obecności rozszerzenia MMX (738)
- 13.3. Środowisko programowania MMX (739)
 - o 13.3.1. Rejestry MMX (739)
 - o 13.3.2. Typy danych MMX (741)
- 13.4. Przeznaczenie instrukcji zestawu MMX (742)
- 13.5. Arytmetyka z nasycaniem a arytmetyka z zawijaniem (743)
- 13.6. Operandy instrukcji MMX (744)
- 13.7. Instrukcje zestawu MMX (746)
 - o 13.7.1. Instrukcje transferu danych (747)
 - o 13.7.2. Instrukcje konwersji (747)
 - o 13.7.3. Arytmetyka operandów spakowanych (752)
 - o 13.7.4. Instrukcje logiczne (755)
 - o 13.7.5. Instrukcje porównań (756)
 - o 13.7.6. Instrukcje przesunięć bitowych (760)
 - o 13.7.7. Instrukcja emms (762)
- 13.8. Model programowania MMX (763)
- 13.9. Źródła informacji dodatkowych (774)

Rozdział 14. Klasy i obiekty (775)

- 14.1. Wprowadzenie (775)
- 14.2. Wstęp do programowania obiektowego (775)
- 14.3. Klasy w języku HLA (779)
- 14.4. Obiekty (782)
- 14.5. Dziedziczenie (784)
- 14.6. Przesłanianie (785)
- 14.7. Metody wirtualne a procedury statyczne (786)
- 14.8. Implementacje metod i procedur klas (788)
- 14.9. Implementacja obiektu (793)
 - o 14.9.1. Tabela metod wirtualnych (796)
 - o 14.9.2. Reprezentacja w pamięci obiektu klasy pochodnej (798)
- 14.10. Konstruktory i inicjalizacja obiektów (802)
 - o 14.10.1. Konstruktor a dynamiczny przydział obiektu (804)
 - o 14.10.2. Konstruktory a dziedziczenie (806)
 - o 14.10.3. Parametry konstruktorów i przeciążanie procedur klas (810)
- 14.11. Destruktory (811)
- 14.12. Łańcuchy `_initialize_` oraz `_finalize_` w języku HLA (812)
- 14.13. Metody abstrakcyjne (818)
- 14.14. Informacja o typie czasu wykonania (RTTI) (822)
- 14.15. Wywołania metod klasy bazowej (824)
- 14.16. Źródła informacji dodatkowych (825)

Rozdział 15. Na styku asemblera i innych języków programowania (827)

- 15.1. Wprowadzenie (827)
- 15.2. Łączenie kodu HLA i kodu asemblera MASM bądź Gas (827)
 - o 15.2.1. Kod asemblera MASM (Gas) rozwijany w kodzie języka HLA (828)
 - o 15.2.2. Konsolidacja modułów MASM (Gas) z modułami HLA (832)
- 15.3. Moduły HLA a programy języka Delphi (Kylix) (837)

- o 15.3.1. Konsolidacja modułów HLA z programami języka Delphi (Kylix) (838)
- o 15.3.2. Zachowywanie wartości rejestrów (842)
- o 15.3.3. Wartości zwracane funkcji (843)
- o 15.3.4. Konwencje wywołań (849)
- o 15.3.5. Przekazywanie argumentów przez wartość i adres, parametry niemodyfikowalne i wyjściowe (854)
- o 15.3.6. Skalarne typy danych w językach HLA i Delphi (Kylix) (856)
- o 15.3.7. Przekazywanie łańcuchów znaków z Delphi do procedury HLA (858)
- o 15.3.8. Przekazywanie rekordów z programu w języku Delphi do kodu HLA (862)
- o 15.3.9. Przekazywanie zbiorów z programu w języku Delphi do kodu HLA (866)
- o 15.3.10. Przekazywanie tablic z programu w języku Delphi do kodu HLA (867)
- o 15.3.11. Odwołania do obiektów programu pisanego w Delphi w kodzie HLA (867)
- 15.4. Moduły HLA a programy języków C i C++ (870)
 - o 15.4.1. Konsolidacja modułów języka HLA z programami języków C i C++ (872)
 - o 15.4.2. Zachowywanie wartości rejestrów (875)
 - o 15.4.3. Wartości funkcji (876)
 - o 15.4.4. Konwencje wywołań (876)
 - o 15.4.5. Tryby przekazywania argumentów (880)
 - o 15.4.6. Odwzorowanie typów skalnych pomiędzy językiem C (C++) a językiem HLA (881)
 - o 15.4.7. Przekazywanie łańcuchów znaków pomiędzy programem w języku C (C++) a modułem HLA (883)
 - o 15.4.8. Przekazywanie rekordów pomiędzy programem w języku C (C++) a modułem HLA (883)
 - o 15.4.9. Przekazywanie tablic pomiędzy programem w języku C (C++) a modułem HLA (886)
- 15.5. Źródła informacji dodatkowych (886)

Dodatek A Tabela kodów ASCII (887)

Dodatek B Instrukcje procesorów 80x86 (891)

Skorowidz (927)