

PODZIĘKOWANIA (15)

1. WSTĘP DO JĘZYKA ASEMBLEROWEGO (17)

- 1.1. Anatomia programu HLA (18)
- 1.2. Uruchamianie pierwszego programu HLA (20)
- 1.3. Podstawowe deklaracje danych programu HLA (22)
- 1.4. Wartości logiczne (24)
- 1.5. Wartości znakowe (25)
- 1.6. Rodzina procesorów 80x86 firmy Intel (25)
- 1.7. Podsystem obsługi pamięci (28)
- 1.8. Podstawowe instrukcje maszynowe (31)
- 1.9. Podstawowe struktury sterujące wykonaniem programu HLA (34)
 - o 1.9.1. Wyrażenia logiczne w instrukcjach HLA (35)
 - o 1.9.2. Instrukcje if..then..elseif..else..endif języka HLA (37)
 - o 1.9.3. Iloczyn, suma i negacja w wyrażeniach logicznych (39)
 - o 1.9.4. Instrukcja while (42)
 - o 1.9.5. Instrukcja for (43)
 - o 1.9.6. Instrukcja repeat (44)
 - o 1.9.7. Instrukcje break oraz breakif (45)
 - o 1.9.8. Instrukcja forever (45)
 - o 1.9.9. Instrukcje try, exception oraz endtry (46)
- 1.10. Biblioteka standardowa języka HLA - wprowadzenie (50)
 - o 1.10.1. Stałe predefiniowane w module stdio (52)
 - o 1.10.2. Standardowe wejście i wyjście programu (53)
 - o 1.10.3. Procedura stdout.newln (54)
 - o 1.10.4. Procedury stdout.putiN (54)
 - o 1.10.5. Procedury stdout.putiNSize (54)
 - o 1.10.6. Procedura stdout.put (56)
 - o 1.10.7. Procedura stdin.getc (58)
 - o 1.10.8. Procedury stdin.getiN (59)
 - o 1.10.9. Procedury stdin.readLn i stdin.flushInput (60)
 - o 1.10.10. Procedura stdin.get (61)
- 1.11. Jeszcze o ochronie wykonania kodu w bloku try..endtry (62)
 - o 1.11.1. Zagnieżdżone bloki try..endtry (63)
 - o 1.11.2. Klauzula unprotected bloku try..endtry (65)
 - o 1.11.3. Klauzula anyexception bloku try..endtry (68)
 - o 1.11.4. Instrukcja try..endtry i rejestry (68)
- 1.12. Język assemblerowy a język HLA (70)
- 1.13. Źródła informacji dodatkowych (71)

2. REPREZENTACJA DANYCH (73)

- 2.1. Systemy liczbowe (74)
 - o 2.1.1. System dziesiętny - przypomnienie (74)
 - o 2.1.2. System dwójkowy (74)
 - o 2.1.3. Formaty liczb dwójkowych (75)
- 2.2. System szesnastkowy (76)
- 2.3. Organizacja danych (79)
 - o 2.3.1. Bity (79)
 - o 2.3.2. Półbajty (79)

- o 2.3.3. Bajty (80)
 - o 2.3.4. Słowa (82)
 - o 2.3.5. Podwójne słowa (83)
 - o 2.3.6. Słowa poczwórne i długie (84)
- 2.4. Operacje arytmetyczne na liczbach dwójkowych i szesnastkowych (85)
- 2.5. Jeszcze o liczbach i ich reprezentacji (86)
- 2.6. Operacje logiczne na bitach (88)
- 2.7. Operacje logiczne na liczbach dwójkowych i ciągach bitów (91)
- 2.8. Liczby ze znakiem i bez znaku (93)
- 2.9. Rozszerzanie znakiem, rozszerzanie zerem, skracanie, przycinanie (98)
- 2.10. Przesunięcia i obroty (102)
- 2.11. Pola bitowe i dane spakowane (107)
- 2.12. Wprowadzenie do arytmetyki zmiennoprzecinkowej (112)
 - o 2.12.1. Formaty zmiennoprzecinkowe przyjęte przez IEEE (116)
 - o 2.12.2. Obsługa liczb zmiennoprzecinkowych w języku HLA (120)
- 2.13. Reprezentacja liczb BCD (124)
- 2.14. Znaki (125)
 - o 2.14.1. Zestaw znaków ASCII (125)
 - o 2.14.2. Obsługa znaków ASCII w języku HLA (129)
- 2.15. Zestaw znaków Unicode (134)
- 2.16. Źródła informacji dodatkowych (134)

3. DOSTĘP DO PAMIĘCI I JEJ ORGANIZACJA (135)

- 3.1. Tryby adresowania procesorów 80x86 (136)
 - o 3.1.1. Adresowanie przez rejestr (136)
 - o 3.1.2. (32-bitowe tryby adresowania procesora 80x86 137)
- 3.2. Organizacja pamięci fazy wykonania (144)
 - o 3.2.1. Obszar kodu (145)
 - o 3.2.2. Obszar zmiennych statycznych (147)
 - o 3.2.3. Obszar niemodyfikowalny (147)
 - o 3.2.4. Obszar danych niezainicjalizowanych (148)
 - o 3.2.5. Atrybut @nostorage (149)
 - o 3.2.6. Sekcja deklaracji var (150)
 - o 3.2.7. Rozmieszczenie sekcji deklaracji danych w programie HLA (151)
- 3.3. Przydział pamięci dla zmiennych w programach HLA (152)
- 3.4. Wyrównanie danych w programach HLA (154)
- 3.5. Wyrażenia adresowe (157)
- 3.6. Koercja typów (159)
- 3.7. Koercja typu rejestru (162)
- 3.8. Pamięć obszaru stosu oraz instrukcje push i pop (164)
 - o 3.8.1. Podstawowa postać instrukcji push (164)
 - o 3.8.2. Podstawowa postać instrukcji pop (166)
 - o 3.8.3. Zachowywanie wartości rejestrów za pomocą instrukcji push i pop (167)
- 3.9. Stos jako kolejka LIFO (168)
 - o 3.9.1. Pozostałe wersje instrukcji obsługi stosu (170)
 - o 3.9.2. Usuwanie danych ze stosu bez ich zdejmowania (172)
- 3.10. Odwoływanie się do danych na stosie bez ich zdejmowania (174)
- 3.11. Dynamiczny przydział pamięci - obszar pamięci serty (176)
- 3.12. Instrukcje inc oraz dec (181)

- 3.13. Pobieranie adresu obiektu (181)
- 3.14. Źródła informacji dodatkowych (182)

4. STAŁE, ZMIENNE I TYPY DANYCH (183)

- 4.1. Kilka dodatkowych instrukcji: intmul, bound i into (184)
- 4.2. Deklaracje stałych i zmiennych w języku HLA (188)
 - o 4.2.1. Typy stałych (192)
 - o 4.2.2. Literały stałych łańcuchowych i znakowych (193)
 - o 4.2.3. Stałe łańcuchowe i napisowe w sekcji const (195)
 - o 4.2.4. Wyrażenia stałowartościowe (197)
 - o 4.2.5. Wielokrotne sekcje const i ich kolejność w programach HLA (200)
 - o 4.2.6. Sekcja val programu HLA (200)
 - o 4.2.7. Modyfikowanie obiektów sekcji val w wybranym miejscu kodu źródłowego programu (201)
- 4.3. Sekcja type programu HLA (202)
- 4.4. Typy wyliczeniowe w języku HLA (203)
- 4.5. Typy wskaźnikowe (204)
 - o 4.5.1. Wskaźniki w języku assemblerowym (206)
 - o 4.5.2. Deklarowanie wskaźników w programach HLA (207)
 - o 4.5.3. Stałe wskaźnikowe i wyrażenia stałych wskaźnikowych (208)
 - o 4.5.4. Zmienne wskaźnikowe a dynamiczny przydział pamięci (209)
 - o 4.5.5. Typowe błędy stosowania wskaźników (209)
- 4.6. Złożone typy danych (214)
- 4.7. Łańcuchy znaków (214)
- 4.8. Łańcuchy w języku HLA (217)
- 4.9. Odwołania do poszczególnych znaków łańcucha (224)
- 4.10. Moduł strings biblioteki standardowej HLA i procedury manipulacji łańcuchami (226)
- 4.11. Konwersje wewnątrzpamięciowe (239)
- 4.12. Zbiory znaków (240)
- 4.13. Implementacja zbiorów znaków w języku HLA (241)
- 4.14. Literały, stałe i wyrażenia zbiorów znaków w języku HLA (243)
- 4.15. Obsługa zbiorów znaków w bibliotece standardowej HLA (245)
- 4.16. Wykorzystywanie zbiorów znaków w programach HLA (249)
- 4.17. Tablice (250)
- 4.18. Deklarowanie tablic w programach HLA (251)
- 4.19. Literały tablicowe (252)
- 4.20. Odwołania do elementów tablicy jednowymiarowej (254)
- 4.21. Porządkowanie tablicy wartości (255)
- 4.22. Tablice wielowymiarowe (257)
 - o 4.22.1. Wierszowy układ elementów tablicy (258)
 - o 4.22.2. Kolumnowy układ elementów tablicy (262)
- 4.23. Przydział pamięci dla tablic wielowymiarowych (263)
- 4.24. Odwołania do elementów tablic wielowymiarowych w języku assemblerowym (266)
- 4.25. Rekordy (struktury) (267)
- 4.26. Stałe rekordowe (270)
- 4.27. Tablice rekordów (271)
- 4.28. Wykorzystanie tablic i rekordów w roli pól rekordów (272)

- 4.29. Wyrównanie pól w ramach rekordu (276)
- 4.30. Wskaźniki na rekordy (278)
- 4.31. Unie (279)
- 4.32. Unie anonimowe (282)
- 4.33. Typy wariantowe (283)
- 4.34. Przestrzenie nazw (284)
- 4.35. Tablice dynamiczne w języku asemblerowym (288)
- 4.36. Źródła informacji dodatkowych (290)

5. PROCEDURY I MODUŁY (291)

- 5.1. Procedury (292)
- 5.2. Zachowywanie stanu systemu (294)
- 5.3. Przedwczesny powrót z procedury (299)
- 5.4. Zmienne lokalne (300)
- 5.5. Symbole lokalne i globalne obiektów innych niż zmienne (306)
- 5.6. Parametry (306)
 - o 5.6.1. Przekazywanie przez wartość (307)
 - o 5.6.2. Przekazywanie przez adres (311)
- 5.7. Funkcje i wartości funkcji (314)
 - o 5.7.1. Zwracanie wartości funkcji (315)
 - o 5.7.2. Złożenie instrukcji języka HLA (316)
 - o 5.7.3. Atrybut @returns procedur języka HLA (319)
- 5.8. Rekurencja (321)
- 5.9. Deklaracje zapowiadające (326)
- 5.10. Deklaracje procedur w HLA 2.0 (327)
- 5.11. Procedury w ujęciu niskopoziomowym - instrukcja call (328)
- 5.12. Rola stosu w procedurach (330)
- 5.13. Rekordy aktywacji (333)
- 5.14. Standardowa sekwencja wejścia do procedury (336)
- 5.15. Standardowa sekwencja wyjścia z procedury (338)
- 5.16. Niskopoziomowa implementacja zmiennych automatycznych (340)
- 5.17. Niskopoziomowa implementacja parametrów procedury (342)
 - o 5.17.1. Przekazywanie argumentów w rejestrach (342)
 - o 5.17.2. Przekazywanie argumentów w kodzie programu (346)
 - o 5.17.3. Przekazywanie argumentów przez stos (348)
- 5.18. Wskaźniki na procedury (373)
- 5.19. Parametry typu procedurowego (377)
- 5.20. Nietypowane parametry wskaźnikowe (378)
- 5.21. Zarządzanie dużymi projektami programistycznymi (379)
- 5.22. Dyrektywa #include (380)
- 5.23. Unikanie wielokrotnego włączania do kodu tego samego pliku (383)
- 5.24. Moduły a atrybut external (384)
 - o 5.24.1. Działanie atrybutu external (389)
 - o 5.24.2. Pliki nagłówkowe w programach HLA (390)
- 5.25. Jeszcze o problemie zaśmiecania przestrzeni nazw (392)
- 5.26. Źródła informacji dodatkowych (395)

6. ARYTMETYKA (397)

- 6.1. Zestaw instrukcji arytmetycznych procesora 80x86 (397)
 - o 6.1.1. Instrukcje mul i imul (398)
 - o 6.1.2. Instrukcje div i idiv (401)
 - o 6.1.3. Instrukcja cmp (404)
 - o 6.1.4. Instrukcje setcc (409)
 - o 6.1.5. Instrukcja test (411)
- 6.2. Wyrażenia arytmetyczne (413)
 - o 6.2.1. Proste przypisania (413)
 - o 6.2.2. Proste wyrażenia (414)
 - o 6.2.3. Wyrażenia złożone (417)
 - o 6.2.4. Operatory przemienne (423)
- 6.3. Wyrażenia logiczne (424)
- 6.4. Idiomy maszynowe a idiomy arytmetyczne (427)
 - o 6.4.1. Mnożenie bez stosowania instrukcji mul, imul i intmul (427)
 - o 6.4.2. Dzielenie bez stosowania instrukcji div i idiv (428)
 - o 6.4.3. Zliczanie modulo n za pośrednictwem instrukcji and (429)
- 6.5. Arytmetyka zmiennoprzecinkowa (430)
 - o 6.5.1. Rejestry jednostki zmiennoprzecinkowej (430)
 - o 6.5.2. Typy danych jednostki zmiennoprzecinkowej (438)
 - o 6.5.3. Zestaw instrukcji jednostki zmiennoprzecinkowej (439)
 - o 6.5.4. Instrukcje przemieszczania danych (439)
 - o 6.5.5. Instrukcje konwersji (442)
 - o 6.5.6. Instrukcje arytmetyczne (445)
 - o 6.5.7. Instrukcje porównań (451)
 - o 6.5.8. Instrukcje ładowania stałych na stos koprocesora (454)
 - o 6.5.9. Instrukcje funkcji przestępnych (455)
 - o 6.5.10. Pozostałe instrukcje jednostki zmiennoprzecinkowej (457)
 - o 6.5.11. Instrukcje operacji całkowitoliczbowych (459)
- 6.6. Tłumaczenie wyrażeń arytmetycznych na kod maszynowy jednostki zmiennoprzecinkowej (459)
 - o 6.6.1. Konwersja notacji wrostkowej do odwrotnej notacji polskiej (461)
 - o 6.6.2. Konwersja odwrotnej notacji polskiej do kodu języka assemblerowego (464)
- 6.7. Obsługa arytmetyki zmiennoprzecinkowej w bibliotece standardowej języka HLA (465)
- 6.8. Źródła informacji dodatkowych (465)

7. NISKOPOZIOMOWE STRUKTURY STERUJĄCE WYKONANIEM PROGRAMU (467)

- 7.1. Struktury sterujące niskiego poziomu (468)
- 7.2. Etykiety instrukcji (468)
- 7.3. Bezwarunkowy skok do instrukcji (instrukcja jmp) (470)
- 7.4. Instrukcje skoku warunkowego (473)
- 7.5. Struktury sterujące "średniego" poziomu - jt i jf (477)
- 7.6. Implementacja popularnych struktur sterujących w języku assemblerowym (477)
- 7.7. Wstęp do podejmowania decyzji (478)
 - o 7.7.1. Instrukcje if..then..else (479)
 - o 7.7.2. Tłumaczenie instrukcji if języka HLA na język assemblerowy (484)

- o 7.7.3. Obliczanie wartości złożonych wyrażeń logicznych - metoda pełnego obliczania wartości wyrażenia (489)
- o 7.7.4. Skrócone obliczanie wyrażeń logicznych (490)
- o 7.7.5. Wady i zalety metod obliczania wartości wyrażeń logicznych (492)
- o 7.7.6. Efektywna implementacja instrukcji if w języku assemblerowym (494)
- o 7.7.7. Instrukcje wyboru (500)
- 7.8. Skoki pośrednie a automaty stanów (511)
- 7.9. Kod spaghetti (514)
- 7.10. Pętle (515)
 - o 7.10.1. Pętle while (515)
 - o 7.10.2. Pętle repeat..until (517)
 - o 7.10.3. Pętle nieskończone (518)
 - o 7.10.4. Pętle for (519)
 - o 7.10.5. Instrukcje break i continue (521)
 - o 7.10.6. Pętle a rejestry (525)
- 7.11. Optymalizacja kodu (526)
 - o 7.11.1. Obliczanie warunku zakończenia pętli na końcu pętli (526)
 - o 7.11.2. Zliczanie licznika pętli wstecz (529)
 - o 7.11.3. Wstępne obliczanie niezmienników pętli (530)
 - o 7.11.4. Rozciąganie pętli (531)
 - o 7.11.5. Zmienne indukcyjne (533)
- 7.12. Mieszane struktury sterujące w języku HLA (534)
- 7.13. Źródła informacji dodatkowych (537)

8. ZAAWANSOWANE OBLICZENIA W JĘZYKU ASEMBLEROWYM (539)

- 8.1. Operacje o zwielokrotnionej precyzji (540)
 - o 8.1.1. Obsługa operacji zwielokrotnionej precyzji w bibliotece standardowej języka HLA (540)
 - o 8.1.2. Dodawanie liczb zwielokrotnionej precyzji (543)
 - o 8.1.3. Odejmowanie liczb zwielokrotnionej precyzji (547)
 - o 8.1.4. Porównanie wartości o zwielokrotnionej precyzji (548)
 - o 8.1.5. Mnożenie operandów zwielokrotnionej precyzji (553)
 - o 8.1.6. Dzielenie wartości zwielokrotnionej precyzji (556)
 - o 8.1.7. Negacja operandów zwielokrotnionej precyzji (566)
 - o 8.1.8. Iloczyn logiczny operandów zwielokrotnionej precyzji (568)
 - o 8.1.9. Suma logiczna operandów zwielokrotnionej precyzji (568)
 - o 8.1.10. Suma wyłączająca operandów zwielokrotnionej precyzji (569)
 - o 8.1.11. Inwersja operandów zwielokrotnionej precyzji (569)
 - o 8.1.12. Przesunięcia bitowe operandów zwielokrotnionej precyzji (570)
 - o 8.1.13. Obroty operandów zwielokrotnionej precyzji (574)
 - o 8.1.14. Operandy zwielokrotnionej precyzji w operacjach wejścia-wyjścia (575)
- 8.2. Manipulowanie operandami różnych rozmiarów (597)
- 8.3. Arytmetyka liczb dziesiętnych (599)
 - o 8.3.1. Literały liczb BCD (601)
 - o 8.3.2. Instrukcje maszynowe daa i das (601)
 - o 8.3.3. Instrukcje maszynowe aaa, aas, aam i aad (603)
 - o 8.3.4. Koprocessor a arytmetyka spakowanych liczb dziesiętnych (605)
- 8.4. Obliczenia w tabelach (607)

- o 8.4.1. Wyszukiwanie w tabeli wartości funkcji (607)
- o 8.4.2. Dopasowywanie dziedziny (613)
- o 8.4.3. Generowanie tabel wartości funkcji (614)
- o 8.4.4. Wydajność odwołań do tabel przeglądowych (618)
- 8.5. Źródła informacji dodatkowych (618)

9. MAKRODEFINICJE I JĘZYK CZASU KOMPILACJI (619)

- 9.1. Język czasu kompilacji - wstęp (619)
- 9.2. Instrukcje #print i #error (621)
- 9.3. Stałe i zmienne czasu kompilacji (623)
- 9.4. Wyrażenia i operatory czasu kompilacji (624)
- 9.5. Funkcje czasu kompilacji (626)
 - o 9.5.1. Funkcje czasu kompilacji - konwersja typów (628)
 - o 9.5.2. Funkcje czasu kompilacji - obliczenia numeryczne (630)
 - o 9.5.3. Funkcje czasu kompilacji - klasyfikacja znaków (630)
 - o 9.5.4. Funkcje czasu kompilacji - manipulacje łańcuchami znaków (631)
 - o 9.5.5. Odwołania do tablicy symboli (632)
 - o 9.5.6. Pozostałe funkcje czasu kompilacji (633)
 - o 9.5.7. Konwersja typu stałych napisowych (634)
- 9.6. Kompilacja warunkowa (635)
- 9.7. Kompilacja wielokrotna (pętle czasu kompilacji) (640)
- 9.8. Makrodefinicje (procedury czasu kompilacji) (644)
 - o 9.8.1. Makrodefinicje standardowe (644)
 - o 9.8.2. Argumenty makrodefinicji (647)
 - o 9.8.3. Symbole lokalne makrodefinicji (654)
 - o 9.8.4. Makrodefinicje jako procedury czasu kompilacji (657)
 - o 9.8.5. Symulowane przeciążanie funkcji (658)
- 9.9. Tworzenie programów czasu kompilacji (664)
 - o 9.9.1. Generowanie tabel wartości funkcji (664)
 - o 9.9.2. Rozciąganie pętli (669)
- 9.10. Stosowanie makrodefinicji w osobnych plikach kodu źródłowego (670)
- 9.11. Źródła informacji dodatkowych (671)

10. MANIPULOWANIE BITAMI (673)

- 10.1. Czym są dane bitowe? (674)
- 10.2. Instrukcje manipulujące bitami (675)
- 10.3. Znacznik przeniesienia w roli akumulatora bitów (683)
- 10.4. Wstawianie i wyodrębnianie łańcuchów bitów (684)
- 10.5. Scalanie zbiorów bitów i rozpraszanie łańcuchów bitowych (688)
- 10.6. Spakowane tablice łańcuchów bitowych (691)
- 10.7. Wyszukiwanie bitów (693)
- 10.8. Zliczanie bitów (696)
- 10.9. Odwracanie łańcucha bitów (699)
- 10.10. Scalanie łańcuchów bitowych (701)
- 10.11. Wyodrębnianie łańcuchów bitów (702)
- 10.12. Wyszukiwanie wzorca bitowego (704)
- 10.13. Moduł bits biblioteki standardowej HLA (705)
- 10.14. Źródła informacji dodatkowych (708)

11. OPERACJE ŁAŃCUCHOWE (709)

- 11.1. Instrukcje łańcuchowe procesorów 80x86 (710)
 - o 11.1.1. Sposób działania instrukcji łańcuchowych (710)
 - o 11.1.2. Przedrostki instrukcji łańcuchowych - repx (711)
 - o 11.1.3. Znacznik kierunku (711)
 - o 11.1.4. Instrukcja movs (714)
 - o 11.1.5. Instrukcja cmps (719)
 - o 11.1.6. Instrukcja scas (723)
 - o 11.1.7. Instrukcja stos (724)
 - o 11.1.8. Instrukcja lods (725)
 - o 11.1.9. Instrukcje lods i stos w złożonych operacjach łańcuchowych (726)
- 11.2. Wydajność instrukcji łańcuchowych procesorów 80x86 (726)
- 11.3. Źródła informacji dodatkowych (727)

12. KLASY I OBIEKTY (729)

- 12.1. Wstęp do programowania obiektowego (730)
- 12.2. Klasy w języku HLA (733)
- 12.3. Obiekty (736)
- 12.4. Dziedziczenie (738)
- 12.5. Przesłanianie (739)
- 12.6. Metody wirtualne a procedury statyczne (740)
- 12.7. Implementacje metod i procedur klas (742)
- 12.8. Implementacja obiektu (747)
 - o 12.8.1. Tabela metod wirtualnych (750)
 - o 12.8.2. Reprezentacja w pamięci obiektu klasy pochodnej (752)
- 12.9. Konstruktory i inicjalizacja obiektów (757)
 - o 12.9.1. Konstruktor a dynamiczny przydział obiektu (759)
 - o 12.9.2. Konstruktory a dziedziczenie (761)
 - o 12.9.3. Parametry konstruktorów i przeciążanie procedur klas (765)
- 12.10. Destruktry (766)
- 12.11. Łańcuchy `_initialize_` oraz `_finalize_` w języku HLA (767)
- 12.12. Metody abstrakcyjne (774)
- 12.13. Informacja o typie czasu wykonania (RTTI) (777)
- 12.14. Wywołania metod klasy bazowej (779)
- 12.15. Źródła informacji dodatkowych (780)

A: TABELA KODÓW ASCII (781)

SKOROWIDZ (785)