

Spis treści

O autorze	11
O recenzentach technicznych	13
Wprowadzenie	15
Rozdział 1. Funkcje podstawowe	23
Dyrektywy preprocesora	25
Makra	26
Kompilacja warunkowa	39
Wskaźniki zmiennych	42
Składnia	43
Operacje arytmetyczne na wskaźnikach zmiennych	45
Wskaźniki ogólne	48
Wielkość wskaźnika	51
Zapomniany wskaźnik	51
Wybrane informacje szczegółowe dotyczące funkcji	54
Anatomia funkcji	54
Waga projektu	55
Zarządzanie stosem	55
Przekazywanie przez wartość kontra przekazywanie przez referencję	56
Wskaźniki funkcji	59
Struktury	61
Dlaczego struktura?	62
Dlaczego typ zdefiniowany przez użytkownika?	62
Jakie możliwości oferuje struktura?	63
Układ pamięci	64
Struktury zagnieżdżone	68
Wskaźniki struktur	69
Podsumowanie	70

Rozdział 2. Od kodu źródłowego do pliku binarnego	73
Proces kompilacji	74
Kompilacja projektu w języku C	76
Etap 1. Uruchomienie preprocesora	83
Etap 2. Kompilacja	84
Etap 3. Uruchomienie asemblera	87
Etap 4. Linkowanie	90
Preprocesor	93
Kompilator	97
Drzewo składniowe	98
Asembler	100
Linker	101
Jak działa linker?	102
Można szukać linkera!	110
Dekorowanie nazw C++	114
Podsumowanie	116
Rozdział 3. Pliki obiektowe	117
Interfejs binarny aplikacji	118
Formaty plików obiektowych	120
Relokowane pliki obiektowe	121
Wykonywalne pliki obiektowe	126
Biblioteki statyczne	130
Biblioteki dynamiczne	138
Ręczne wczytywanie bibliotek współdzielonych	143
Podsumowanie	145
Rozdział 4. Struktura pamięci procesu	147
Układ pamięci procesu	148
Określanie struktury pamięci	149
Analiza statycznego układu pamięci	150
Segment BSS	152
Segment data	154
Segment text	158
Analiza dynamicznego układu pamięci	160
Mapowanie pamięci	161
Segment stack	165
Segment sterty	167
Podsumowanie	170
Rozdział 5. Stos i sterta	173
Stos	174
Analizowanie stosu	175
Kwestie związane z używaniem pamięci stosu	182
Sterna	185
Alokacja i zwalnianie pamięci na sterzie	187
Reguły dotyczące pamięci sterty	196

Zarządzanie pamięcią w ograniczonym środowisku	199
Środowiska o ograniczonej ilości pamięci	200
Środowisko charakteryzujące się większą wydajnością działania	202
Podsumowanie	208
Rozdział 6. Programowanie zorientowane obiektowo i hermetyzacja	211
Myślenie w sposób zorientowany obiektowo	214
Koncepcje myślowe	215
Mapowanie idei w głowie i modele obiektowe	216
Obiekty nie znajdują się w kodzie	218
Atrybuty obiektu	219
Domena	220
Relacje między obiektami	221
Operacje zorientowane obiektowo	222
Obiekt ma zdefiniowane zachowanie	224
Dlaczego język C nie jest zorientowany obiektowo?	225
Hermetyzacja	226
Hermetyzacja atrybutu	226
Hermetyzacja zachowania	229
Ukrywanie informacji	239
Podsumowanie	246
Rozdział 7. Kompozycja i agregacja	247
Związki między klasami	247
Obiekt kontra klasa	248
Kompozycja	250
Agregacja	256
Podsumowanie	263
Rozdział 8. Dziedziczenie i polimorfizm	265
Dziedziczenie	266
Natura dziedziczenia	267
Polimorfizm	281
Czym jest polimorfizm?	281
Do czego jest potrzebny polimorfizm?	284
Jak w języku C zaimplementować zachowanie polimorficzne?	285
Podsumowanie	292
Rozdział 9. Abstrakcja i programowanie zorientowane obiektowo w C++	293
Abstrakcja	294
Zorientowane obiektowo konstrukcje w C++	297
Hermetyzacja	298
Dziedziczenie	301
Polimorfizm	306
Klasa abstrakcyjna	309
Podsumowanie	311

Rozdział 10. UNIX — historia i architektura	313
Historia systemu UNIX	314
Systemy operacyjne Multics i UNIX	314
Języki BCPL i B	316
Droga do powstania języka C	317
Architektura systemu UNIX	318
Filozofia systemu UNIX	319
Warstwy systemu UNIX	320
Interfejs powłoki dla aplikacji użytkownika	323
Interfejs jądra do warstwy powłoki	327
Jądro	333
Sprzęt	337
Podsumowanie	339
Rozdział 11. Jądro i wywołania systemowe	341
Wywołania systemowe	342
Wywołania systemowe pod mikroskopem	342
Pominięcie standardu C — bezpośrednie wykonanie wywołania systemowego	344
Wewnątrz funkcji wywołania systemowego	346
Dodanie nowego wywołania systemowego do systemu Linux	348
Jądro systemu UNIX	361
Jądro monolityczne kontra mikrojądro	362
Linux	364
Moduły jądra	364
Podsumowanie	370
Rozdział 12. Najnowsza wersja C	371
C11	372
Określenie obsługiwanej wersji standardu języka C	372
Usunięcie funkcji gets()	374
Zmiany wprowadzone w funkcji fopen()	374
Funkcje sprawdzające granice bufora	376
Funkcja niekończąca działania	377
Makra typu generycznego	378
Unicode	378
Unie i struktury anonimowe	384
Wielowątkowość	386
Słowo o standardzie C18	386
Podsumowanie	386
Rozdział 13. Współbieżność	389
Wprowadzenie do współbieżności	390
Równoległość	391
Współbieżność	392
Jednostka zarządcy zadań	393
Procesy i wątki	395
Ograniczenie typu „zachodzi wcześniej”	396

Kiedy należy używać współbieżności	398
Stan współdzielony	405
Podsumowanie	410
Rozdział 14. Synchronizacja	411
Problemy związane ze współbieżnością	412
Wrodzone problemy ze współbieżnością	413
Problemy pojawiające się po synchronizacji	423
Techniki synchronizacji	424
Techniki busy-waiting i spinlock	425
Mechanizm uśpienia-powiadomienia	428
Semaforey i muteksy	431
Wiele jednostek procesora	436
Blokada typu spinlock	441
Zmienne warunkowe	442
Współbieżność w standardzie POSIX	444
Obsługa współbieżności przez jądro	445
Wieloprocusowość	447
Wielowątkowość	450
Podsumowanie	451
Rozdział 15. Wykonywanie wątków	453
Wątki	454
Wątki POSIX	457
Tworzenie wątków POSIX	458
Przykład stanu wyjściu	463
Przykład wyjściu danych	471
Podsumowanie	474
Rozdział 16. Synchronizacja wątków	477
Kontrola współbieżności w standardzie POSIX	478
Muteksy POSIX	478
Zmienne warunkowe POSIX	481
Bariery POSIX	485
Semaforey POSIX	487
Wątki POSIX i pamięć	495
Pamięć stosu	495
Pamięć sterty	500
Widoczność pamięci	504
Podsumowanie	506
Rozdział 17. Wykonywanie procesów	507
API wykonywania procesu	507
Tworzenie procesu	510
Wykonywanie procesu	515
Porównanie tworzenia procesu i wykonywania procesu	517
Procedura wykonania procesu	518

Stan współdzielony	519
Techniki współdzielenia	520
Pamięć współdzielona w standardzie POSIX	522
System plików	531
Wielowątkowość kontra wieloprocusowość	534
Wielowątkowość	534
Wieloprocusowość w pojedynczym komputerze	535
Wieloprocusowość rozproszona	535
Podsumowanie	536
 Rozdział 18. Synchronizacja procesów	 537
Kontrola współbieżności w pojedynczym hoście	538
Nazwane semaforey POSIX	539
Nazwane muteksy	543
Przykład pierwszy	543
Przykład drugi	547
Nazwane zmienne warunkowe	556
Etap 1. Klasa pamięci współdzielonej	557
Etap 2. Klasa współdzielonego licznika w postaci 32-bitowej liczby całkowitej	560
Etap 3. Klasa muteksu współdzielonego	562
Etap 4. Klasa współdzielonej zmiennej warunkowej	565
Etap 5. Logika funkcji main()	568
Kontrola współbieżności rozproszonej	572
Podsumowanie	574
 Rozdział 19. Gniazda i IPC w pojedynczym hoście	 577
Techniki IPC	578
Protokół komunikacyjny	580
Cechy charakterystyczne protokołu	582
Sekwencyjność	584
Komunikacja w pojedynczym hoście	584
Deskryptory plików	585
Sygnały POSIX	585
Potoki POSIX	589
Kolejka komunikatów POSIX	592
Gniazda domeny systemu UNIX	594
Wprowadzenie do programowania gniazd	595
Sieci komputerowe	595
Na czym polega programowanie gniazda?	607
Podsumowanie	614
 Rozdział 20. Programowanie oparte na gniazdach	 615
Podsumowanie informacji o programowaniu gniazd	616
Projekt kalkulatora	618
Hierarchia kodu źródłowego	619
Zbudowanie projektu	623
Uruchomienie projektu	623

Protokół aplikacji	624
Biblioteka serializacji i deserializacji	627
Usługa kalkulatora	632
Gniazda domeny systemu UNIX	634
Serwer strumienia UDS	634
Klient strumienia UDS	641
Serwer datagramu nasłuchujący na gnieździe UDS	644
Klient datagramu używającego gniazda UDS	648
Gniazda sieciowe	650
Serwer TCP	650
Klient TCP	652
Serwer UDP	653
Klient UDP	654
Podsumowanie	655
Rozdział 21. Integracja z innymi językami programowania	657
Dlaczego integracja w ogóle jest możliwa?	658
Pobranie niezbędnych materiałów	659
Biblioteka stosu	660
Integracja z C++	666
Dekorowanie nazw w C++	667
Kod C++	669
Integracja z Javą	673
Utworzenie kodu w Javie	674
Przygotowanie części natywnej	679
Integracja z Pythonem	686
Integracja z Go	689
Podsumowanie	692
Rozdział 22. Testy jednostkowe i debugowanie	695
Testowanie oprogramowania	696
Poziomy testowania	697
Testy jednostkowe	698
Dublerzy używane podczas testów	706
Testowanie komponentu	707
Biblioteki testowania kodu w C	709
Framework CMocka	710
Google Test	718
Debugowanie	722
Kategorie błędów	724
Debugery	724
Narzędzia profilowania pamięci	726
Debugery wątków	728
Narzędzia do profilowania wydajności działania	729
Podsumowanie	730

Rozdział 23. Systemy kompilacji	731
Czym jest system kompilacji?	732
Make	733
CMake — to nie jest system kompilacji!	740
Ninja	745
Bazel	747
Porównanie systemów kompilacji	750
Podsumowanie	750
Epilog	751