

O Autorach (11)

Wprowadzenie (13)

Część I Z Delphi do Kyliksa (17)

Rozdział 1. Porównanie środowisk (19)

- Różnice pomiędzy środowiskami (19)
 - o Różnice systemowe (20)
 - o Różnice występujące w środowisku programowania (22)
 - o Open Source (23)
- Okna w Linuksie (24)
 - o System X Window (24)
 - o Menedżer okien (25)
- Gdzie znaleźć więcej informacji (27)

Rozdział 2. Krótki wykład na temat Linuksa (29)

- Pliki i katalogi (29)
 - o Nazwy plików i katalogów (30)
 - o Łączy do plików (32)
 - o Własność plików i katalogów (33)
 - o Do czego służą te katalogi? (35)
- Techniki programowania Linuksa (36)
 - o Koncepcja wywołań systemowych (37)
 - o Procesy i sygnały (38)
 - o Programowanie wątków (39)
 - o Komunikacja pomiędzy procesami (40)
- System plików /proc (43)
- Gdzie znaleźć więcej informacji (44)

Rozdział 3. Korzystanie z bibliotek (45)

- Tworzenie obiektu wspólnego i korzystanie z niego (46)
 - o Tworzenie grupy projektów (46)
 - o Biblioteka Hello (47)
 - o Wywoływanie funkcji bibliotecznych (48)
 - o Udostępnianie bibliotek (49)
- Konwencje wspólnych bibliotek (50)
 - o Konwencje nazywania bibliotek (50)
 - o Gdzie umieścić biblioteki? (52)
 - o Gdzie Linux szuka bibliotek? (52)
 - o Kylix a zgodność z konwencją bibliotek (53)
 - o Zabawa z nazwami funkcji (54)
- Dynamiczne ładowanie bibliotek (55)
 - o Inicjalizacja i końcowe działania bibliotek (58)
 - o Gdzie szuka bibliotek funkcja dlopen (58)
- Przenoszenie bibliotek pomiędzy platformami (58)
 - o Konwencje wywołań (59)
 - o Nazwa biblioteki (59)
 - o Typ uchwytu biblioteki (60)
 - o Moduł ShareMem (60)

- o Kod startowy biblioteki (60)
 - o Moduł interfejsu dla różnych platform (61)
- Gdzie znaleźć więcej informacji (63)

Rozdział 4. Podstawy C dla programistów Kyliksa (65)

- Podręcznik C dla programisty Pascala (66)
 - o Typy danych C i Pascala (66)
 - o Zmienne, struktury, unie oraz typy (68)
 - o Operatory języka C (70)
 - o Instrukcje sterujące i pętle (73)
 - o Wskaźniki (75)
 - o Funkcje i "procedury" (78)
 - o Pozostałe zagadnienia (79)
 - o Elementy C++ (81)
- Używanie bibliotek C w Kyliksie (83)
 - o Tworzenie programów za pomocą polecenia make (84)
 - o Tworzenie modułu interfejsu (86)
- Gdzie znaleźć więcej informacji (92)

Część II Przykłady programowania Kyliksa (95)

Rozdział 5. Kontrolowanie procesów (97)

- 5.1 Zastępowanie bieżącego procesu nowym programem (98)
 - o Przykład: program execTest (100)
- 5.2 Uruchamianie procesu potomnego (104)
 - o Przykład: tworzenie procesu potomnego z programu konsolowego (105)
 - o Przykład: tworzenie procesu potomnego w aplikacji graficznego interfejsu użytkownika (106)
- 5.3 Uruchamianie programu i czekanie na zakończenie jego działania (107)
 - o Przykład: program forkWait (107)
- 5.4 Tworzenie procesu działającego w tle (109)
 - o Przykład: program bgTest (109)
- 5.5 Niszczenie procesu (110)
 - o Przykład: program CrashTestDummy (110)
 - o Przykład: moduł ProcStuff (112)
 - o Przykład: program KillerApp (116)
- 5.6 Ustalanie priorytetów procesów (118)
 - o Przykład: program priTest (120)
- 5.7 Zmniejszanie obciążenia systemu (121)
 - o Przykład: program Sleeper (122)
- 5.8 Uzyskiwanie informacji na temat identyfikatora procesu oraz użytkownika (124)
 - o Przykład: program GetPID (125)
- 5.9 Uzyskiwanie szczegółowych informacji na temat procesu (127)
 - o Przykład: wyliczanie procesów (128)
 - o Przykład: uzyskiwanie linii polecenia procesu (129)
 - o Przykład: uzyskiwanie środowiska procesu (131)
 - o Przykład: uzyskiwanie pliku wykonywalnego, katalogu głównego oraz katalogu roboczego procesu (131)

- o Przykład: uzyskiwanie informacji na temat plików otwartych przez proces (132)
 - o Przykład: uzyskiwanie szczegółowych informacji na temat procesu (134)
- 5.10 Ograniczanie działania procesu do pojedynczej instancji (136)
 - o Przykład: program OneInst (137)
- 5.11 Ustalanie harmonogramu działania procesu (140)
 - o Przykład: skrypt startdummy (141)
 - o Przykład: program RunAt (141)
 - o Przykład: program RunScheduled (144)
- 5.12 Uruchamianie procesu przez superużytkownika (149)

Rozdział 6. Obsługa komunikacji między procesami (151)

- 6.1 Podstawowa komunikacja przeprowadzana z użyciem sygnałów (152)
 - o Przykład: program SigSender (154)
- 6.2 Uzyskiwanie opisów poprawnych sygnałów systemowych (157)
 - o Przykład: program SigNames (157)
- 6.3 Tworzenie procedur obsługi sygnałów (158)
 - o Przykład: program HndSender (158)
 - o Funkcja sigaction oraz obiekt TSigAction (159)
 - o Zarządzanie zestawami sygnałów (161)
 - o Przykład: program HndRecvr (162)
- 6.4 Zabezpieczanie przed procesami zombie (165)
 - o Przykład: program KillAllZombies (165)
- 6.5 Komunikacja z aplikacjami konsoli poprzez potoki (168)
 - o Przykład: program PipeRead (169)
 - o Przykład: program RunAt (jeszcze raz) (171)
- 6.6 Przekazywanie danych pomiędzy potomnym a nadrzędnym procesem GUI (172)
 - o Przykład: PipeParent (173)
 - o Przykład: PipeChild (176)
- 6.7 Przekazywanie danych pomiędzy niezależnymi procesami GUI (178)
 - o Przykład: FIFOsender (179)
 - o Przykład: FIFORecvr (181)
- 6.8 Koordynowanie procesów za pomocą semaforów (183)
 - o Praca z obiektami IPC (184)
 - o Semaforey (185)
 - o Funkcje semaforów (185)
 - o Tworzenie i otwieranie zestawów semaforów (186)
 - o Kontrolowanie semaforów za pomocą funkcji semctl (187)
 - o Przykład: program OneAtATime (192)
- 6.9 Wysoko wydajna komunikacja przeprowadzona z użyciem pamięci wspólnej (195)
 - o Funkcje pamięci wspólnej (195)
 - o Tworzenie i otwieranie obiektów pamięci wspólnej (196)
 - o Kontrolowanie obiektów pamięci wspólnej (197)
 - o Przykład: program shmTalk (198)
- 6.10 Komunikowanie się poprzez kolejki komunikatów (202)
 - o Funkcje kolejek komunikatów (202)
 - o Tworzenie i otwieranie kolejek komunikatów (203)
 - o Kontrolowanie kolejek komunikatów (203)
 - o Wysyłanie komunikatów (205)

- o Odbieranie komunikatów (206)
- o Praca z różnymi rodzajami komunikatów (207)
- o Kolejki komunikatów i wskaźniki (209)
- o Przykład: prosty dziennik debugowania (209)

Rozdział 7. Korzystanie z systemu plików (213)

- 7.1 Sprawdzanie uprawnień dla pliku (213)
 - o Przykład: program GetPermissions (214)
- 7.2 Definiowanie uprawnień dla pliku (216)
 - o Przykład: program SetPermissions (216)
- 7.3 Implementacja blokowania plików danych na poziomie plików (218)
 - o Przykład: program LockFile (219)
- 7.4 Implementacja blokowania plików danych na poziomie rekordów (222)
 - o Polecenia blokowania pliku (222)
 - o Przypadek hipotetyczny (224)
 - o Przykład: program LockWriter (224)
 - o Przykład: program LockReader (228)
- 7.5 Uzyskiwanie atrybutów i informacji na temat pliku (234)
 - o Przykład: program FileInfo (236)
- 7.6 Montowanie innych systemów plików (239)
 - o Przykład: program MountTool (242)

Rozdział 8. Pomoc online (247)

- Architektura systemu pomocy (247)
- 8.1 Tworzenie prostej przeglądarki pomocy (248)
 - o Przykład: klasa TSimpleHelpViewer (250)
- 8.2 Dodawanie pomocy do aplikacji (255)
 - o Przykład: dodawanie pomocy do aplikacji (257)
- 8.3 Współpraca z zewnętrznym systemem pomocy (258)
 - o Formularze SimpleHelp (259)
 - o Format plików pomocy (259)
 - o Przykład: implementowanie systemu pomocy SimpleHelp (260)
 - o Przyszłość pomocy w Kyliksie (267)

Rozdział 9. Szuflada (269)

- 9.1 Uzyskiwanie listy zalogowanych użytkowników (270)
 - o Przykład: program LogUser (272)
- 9.2 Sprawdzanie nie odczytanej poczty (274)
 - o Przykład: program CheckMail (275)
- 9.3 Wysyłanie poczty pod adresy lokalne (283)
 - o Przykład: program PipeMail (284)
- 9.4 Użycie programu sendmail z aplikacji (287)
 - o Wysyłanie za pomocą programu sendmail (287)
 - o Przykład: program FileMail (288)
- 9.5 Posługiwanie się ósemkowymi maskami uprawnień (292)
 - o Przykład: program OctalConv (292)
- 9.6 Uruchamianie programu jako superużytkownik (ponownie) (296)

- o Dziedziczenie i środowisko (297)
 - o Przykład: zmiana konfiguracji użytkownika (299)
 - o Przykład: skrypt runsu1 (299)
 - o Przykład: skrypt runsu2 (300)
- 9.7 Wykorzystanie dzienników systemowych do debugowania (301)
 - o Systemowe procedury dzienników (302)
 - o Ukryty demon (303)
 - o Przykład: program SysLog (305)

Skorowidz (309)