- Linux: Programowanie Systemowe
- Przedmowa
- Wstęp
    - Przyjęte założenia, dotyczące odbiorców książki
    - Zawartość książki
    - Wersje uwzględnione w tej książce
    - Użyte konwencje
    - Używanie przykładowych kodów
    - Podziękowania
- 1. Wprowadzenie podstawowe pojęcia
    - Programowanie systemowe
        - Dlaczego warto uczyć się programowania systemowego?
        - Kamienie węgielne programowania systemowego
        - Funkcje systemowe
            - Wywoływanie funkcji systemowych
        - Biblioteka języka C
        - Kompilator języka C
    - API i ABI
        - API
        - ABI
    - Standardy
        - Historia POSIX oraz SUS
        - Standardy języka C
        - Linux i standardy
        - Książka i standardy
    - Pojęcia dotyczące programowania w Linuksie
        - Pliki i system plików
            - Pliki zwykłe
            - Katalogi i dowiązania
            - Dowiązania twarde
            - Dowiązania symboliczne
            - Pliki specjalne
            - Systemy plików i przestrzenie nazw
        - Procesy
            - Wątki
            - Hierarchia procesów
        - Użytkownicy i grupy
        - Uprawnienia
        - Sygnały
        - Komunikacja międzyprocesowa
        - Pliki nagłówkowe
        - Obsługa błędów
    - Początek programowania systemowego
- 2. Plikowe operacje wejścia i wyjścia
    - Otwieranie plików
        - Funkcja systemowa open()
            - Znaczniki w funkcji open()
        - Właściciele nowych plików
        - Uprawnienia nowych plików
        - Funkcja creat()

        - Wartości zwracane i kody błędów
    - Czytanie z pliku przy użyciu funkcji read()
        - Wartości zwracane
        - Czytanie wszystkich bajtów
        - Odczyty nieblokujące
        - Inne wartości błędów
        - Ograniczenia rozmiaru dla funkcji read()
    - Pisanie za pomocą funkcji write()
        - Zapisy częściowe
        - Tryb dopisywania
        - Zapisy nieblokujące
        - Inne kody błędów
        - Ograniczenia rozmiaru dla funkcji write()
        - Sposób działania funkcji write()
    - Zsynchronizowane operacje wejścia i wyjścia
        - Funkcje fsync() i fdatasync()
            - Wartości zwracane i kody błędów
        - Funkcja sync()
        - Znacznik O_SYNC
        - Znaczniki O_DSYNC i O_RSYNC
    - Bezpośrednie operacje wejścia i wyjścia
    - Zamykanie plików
        - Kody błędów
    - Szukanie za pomocą funkcji lseek()
        - Szukanie poza końcem pliku
        - Kody błędów
        - Ograniczenia
    - Odczyty i zapisy pozycyjne
        - Kody błędów
    - Obcinanie plików
    - Zwielokrotnione operacje wejścia i wyjścia
        - Funkcja select()
            - Wartości powrotu oraz kody błędów
            - Przykład użycia funkcji select()
            - Przenośny sposób wstrzymania wykonania aplikacji za pomocą funkcji select()
            - Funkcja pselect()
        - Funkcja poll()
            - Wartości powrotu oraz kody błędów
            - Przykład użycia funkcji poll()
            - Funkcja ppoll()
        - Porównanie funkcji poll() i select()
    - Organizacja wewnętrzna jądra
        - Wirtualny system plików
        - Bufor stron
        - Opóźniony zapis stron
    - Zakończenie
- 3. Buforowane operacje wejścia i wyjścia
    - Operacje wejścia i wyjścia buforowane w przestrzeni użytkownika
        - Rozmiar bloku

- Typowe operacje wejścia i wyjścia
  - Wskaźniki do plików
- Otwieranie plików
  - Tryby
- Otwieranie strumienia poprzez deskryptor pliku
- Zamykanie strumieni
  - Zamykanie wszystkich strumieni
- Czytanie ze strumienia
  - Czytanie pojedynczego znaku
    - Wycofywanie znaku
  - Czytanie całego wiersza
    - Czytanie dowolnych łańcuchów
  - Czytanie danych binarnych
- Pisanie do strumienia
  - Zapisywanie pojedynczego znaku
  - Zapisywanie łańcucha znaków
  - Zapisywanie danych binarnych
- Przykładowy program używający buforowanych operacji wejścia i wyjścia
- Szukanie w strumieniu
  - Otrzymywanie informacji o aktualnym położeniu w strumieniu
- Opróżnianie strumienia
- Błędy i koniec pliku
- Otrzymywanie skojarzonego deskryptora pliku
- Parametry buforowania
- Bezpieczeństwo wątków
  - Nieautomatyczne blokowanie plików
  - Nieblokowane operacje na strumieniu
- Krytyczna analiza biblioteki typowych operacji wejścia i wyjścia
- Zakończenie

- 4. Zaawansowane operacje plikowe wejścia i wyjścia
  - Rozproszone operacje wejścia i wyjścia
    - Funkcje readv() i writev()
      - Wartości powrotne
      - Przykład użycia funkcji writev()
      - Przykład użycia funkcji readv()
      - Implementacja
  - Odpytywanie zdarzeń
    - Tworzenie nowego egzemplarza interfejsu odpytywania zdarzeń
    - Sterowanie działaniem interfejsu odpytywania zdarzeń
    - Oczekiwanie na zdarzenie w interfejsie odpytywania zdarzeń
    - Zdarzenia przełączane zboczem a zdarzenia przełączane poziomem
  - Odwzorowywanie plików w pamięci
    - Funkcja mmap()
      - Rozmiar strony
      - Wartości powrotne i kody błędów
      - Dodatkowe sygnały
    - Funkcja munmap()
    - Przykład odwzorowania w pamięci
    - Zalety używania funkcji mmap()
    - Wady używania funkcji mmap()

        - Zmiana rozmiaru odwzorowania
            - Wartości powrotne i kody błędów
        - Zmiana uprawnień odwzorowania
            - Wartości powrotne i kody błędów
        - Synchronizacja odwzorowanego pliku
            - Wartości powrotne i kody błędów
        - Dostarczanie porad dotyczących odwzorowania w pamięci
            - Wartości powrotne i kody błędów
    - Porady dla standardowych operacji plikowych wejścia i wyjścia
        - Funkcja systemowa posix_fadvise()
            - Wartości powrotne i kody błędów
        - Funkcja systemowa readahead()
            - Wartości powrotne i kody błędów
        - Porada jest tania
    - Operacje zsynchronizowane, synchroniczne i asynchroniczne
        - Asynchroniczne operacje wejścia i wyjścia
    - Zarządcy operacji wejścia i wyjścia oraz wydajność operacji wejścia i wyjścia
        - Adresowanie dysku
        - Działanie zarządcy operacji wejścia i wyjścia
        - Wspomaganie odczytów
            - Zarządca z terminem nieprzekraczalnym
            - Zarządca przewidujący
            - Zarządca ze sprawiedliwym szeregowaniem
            - Zarządca niesortujący
        - Wybór i konfiguracja zarządcy operacji wejścia i wyjścia
        - Optymalizowanie wydajności operacji wejścia i wyjścia
            - Szeregowanie operacji wejścia i wyjścia w przestrzeni użytkownika
                - Sortowanie wg ścieżki
                - Sortowanie wg numeru i-węzła.
                - Sortowanie wg numeru fizycznego bloku
    - Zakończenie
- 5. Zarządzanie procesami
    - Programy, procesy i wątki
    - Identyfikator procesu
        - Przydział identyfikatorów procesów
        - Hierarchia procesów
        - Typ pid_t
        - Otrzymywanie identyfikatora procesu oraz identyfikatora procesu rodzicielskiego
    - Uruchamianie nowego procesu
        - Rodzina funkcji exec
            - Pozostałe elementy grupy funkcji exec
            - Kody błędów
        - Funkcja systemowa fork()
            - Kopiowanie podczas zapisu
            - Funkcja vfork()
    - Zakończenie procesu
        - Inne sposoby na zakończenie procesu
        - Funkcja atexit()

        - Funkcja on_exit()
        - Sygnał SIGCHLD
    - Oczekiwanie na zakończone procesy potomka
        - Oczekiwanie na określony proces
        - Jeszcze wszechstronniejsza funkcja oczekiwania
        - BSD wkracza do akcji: funkcje wait3() i wait4()
        - Uruchamianie i oczekiwanie na nowy proces
        - Procesy zombie
    - Użytkownicy i grupy
        - Rzeczywiste, efektywne oraz zapisane identyfikatory użytkownika i grupy
        - Zmiana rzeczywistego lub zapisanego identyfikatora dla użytkownika lub grupy
        - Zmiana efektywnego identyfikatora dla użytkownika lub grupy
        - Zmiana identyfikatora dla użytkownika lub grupy w wersji BSD
        - Zmiana identyfikatora dla użytkownika lub grupy w wersji HP-UX
        - Zalecane modyfikacje identyfikatorów użytkownika i grupy
        - Wsparcie dla zapisanych identyfikatorów użytkownika
        - Otrzymywanie identyfikatorów użytkownika i grupy
    - Grupy sesji i procesów
        - Funkcje systemowe do obsługi sesji
        - Funkcje systemowe do obsługi grup procesów
        - Przestarzałe funkcje do obsługi grupy procesów
    - Demony
    - Zakończenie
- 6. Zaawansowane zarządzanie procesami
    - Szeregowanie procesów
        - Przedziały czasowe
        - Procesy związane z wejściem i wyjściem a procesy związane z procesorem
        - Szeregowanie z wywłaszczaniem
    - Completely Fair Scheduler
    - Udostępnianie czasu procesora
        - Prawidłowe sposoby użycia sched_yield()
    - Priorytety procesu
        - nice()
        - getpriority() i setpriority()
        - Priorytety wejścia i wyjścia
    - Wiązanie procesów do konkretnego procesora
        - sched_getaffinity() i sched_setaffinity()
    - Systemy czasu rzeczywistego
        - Systemy ścisłego oraz zwykłego czasu rzeczywistego
        - Opóźnienie, rozsynchronizowanie oraz parametry graniczne
        - Obsługa czasu rzeczywistego przez system Linux
        - Linuksowe strategie szeregowania i ustalania priorytetów
            - Strategia FIFO (first in, first out)
            - Strategia cykliczna
            - Strategia zwykła
            - Strategia szeregowania wsadowego
            - Ustalanie strategii szeregowania dla systemu Linux

            - Kody błędów
        - Ustawianie parametrów szeregowania
            - Kody błędów
            - Określanie zakresu poprawnych priorytetów
        - sched_rr_get_interval()
            - Kody błędu
        - Środki ostrożności przy pracy z procesami czasu rzeczywistego
        - Determinizm
            - Wcześniejsze zapisywanie danych oraz blokowanie pamięci
            - Wiązanie do procesora a procesy czasu rzeczywistego
    - Ograniczenia zasobów systemowych
        - Ograniczenia
            - Ograniczenia domyślne
        - Ustawianie i odczytywanie ograniczeń
            - Kody błędów
- 7. Wątkowość
    - Binaria, procesy i wątki
    - Wielowątkowość
        - Koszty wielowątkowości
        - Alternatywy dla wielowątkowości
    - Modele wątkowości
        - Wątkowość na poziomie użytkownika
        - Wątkowość mieszana
        - Współprogramy i włókna
    - Wzorce wątkowości
        - Wątkowość thread-per-connection
        - Wątkowość sterowana zdarzeniami
    - Współbieżność, równoległość i wyścigi
        - Sytuacje wyścigów
            - Przykłady wyścigów ze świata rzeczywistego
    - Synchronizacja
        - Muteksy
        - Zakleszczenia
            - Unikanie zakleszczeń
    - Standard Pthreads
        - Implementacje wątkowości w Linuksie
        - Interfejs programistyczny dla standardu Pthreads
        - Konsolidowanie implementacji Pthreads
        - Tworzenie wątków
        - Identyfikatory wątków
            - Porównywanie identyfikatorów wątków
        - Kończenie wątków
            - Zakończenie samego siebie
            - Zakończenie innych wątków
        - Łączenie i odłączanie wątków
            - Łączenie wątków
            - Odłączanie wątków
        - Przykład wątkowości
        - Muteksy standardu Pthreads
            - Inicjalizowanie muteksów

            - Blokowanie muteksów
            - Odblokowywanie muteksów
            - Przykład użycia muteksu
    - Dalsze zdobywanie wiedzy
- 8. Zarządzanie plikami i katalogami
    - Pliki i ich metadane
        - Rodzina funkcji stat
        - Uprawnienia
        - Prawa własności
        - Atrybuty rozszerzone
            - Klucze i wartości
            - Przestrzenie nazw dla atrybutów rozszerzonych
        - Operacje dla atrybutów rozszerzonych
            - Odczytywanie atrybutu rozszerzonego
            - Ustawianie atrybutu rozszerzonego
            - Wyświetlanie atrybutów rozszerzonych dla pliku
            - Usuwanie atrybutu rozszerzonego
    - Katalogi
        - Aktualny katalog roboczy
            - Odczytywanie aktualnego katalogu roboczego
            - Zmiana aktualnego katalogu roboczego
        - Tworzenie katalogów
        - Usuwanie katalogów
        - Odczytywanie zawartości katalogu
            - Czytanie ze strumienia katalogu
            - Zamykanie strumienia katalogu
            - Funkcje systemowe służące do odczytywania zawartości katalogu
    - Dowiązania
        - Dowiązania twarde
        - Dowiązania symboliczne
        - Usuwanie elementów z systemu plików
    - Kopiowanie i przenoszenie plików
        - Kopiowanie
        - Przenoszenie
    - Węzły urządzeń
        - Specjalne węzły urządzeń
        - Generator liczb losowych
    - Komunikacja poza kolejką
    - Śledzenie zdarzeń związanych z plikami
        - Inicjalizacja interfejsu inotify
        - Elementy obserwowane
            - Dodawanie nowego elementu obserwowanego
            - Maska elementu obserwowanego
        - Zdarzenia interfejsu inotify
            - Odczytywanie zdarzeń inotify
            - Zaawansowane zdarzenia inotify
            - Łączenie zdarzeń przenoszenia
        - Zaawansowane opcje obserwowania
        - Usuwanie elementu obserwowanego z interfejsu inotify

        - Otrzymywanie rozmiaru kolejki zdarzeń
        - Usuwanie egzemplarza interfejsu inotify
- 9. Zarządzanie pamięcią
    - Przestrzeń adresowa procesu
        - Strony i stronicowanie
            - Współdzielenie i kopiowanie podczas zapisu
        - Regiony pamięci
    - Przydzielanie pamięci dynamicznej
        - Przydzielanie pamięci dla tablic
        - Zmiana wielkości obszaru przydzielonej pamięci
        - Zwalnianie pamięci dynamicznej
        - Wyrównanie
            - Przydzielanie pamięci wyrównanej
            - Inne zagadnienia związane z wyrównaniem
    - Zarządzanie segmentem danych
    - Anonimowe odwzorowania w pamięci
        - Tworzenie anonimowych odwzorowań w pamięci
        - Odwzorowanie pliku /dev/zero
    - Zaawansowane operacje przydziału pamięci
        - Dokładne dostrajanie przy użyciu funkcji malloc_usable_size() oraz malloc_trim()
    - Uruchamianie programów używających systemu przydzielania pamięci
        - Otrzymywanie danych statystycznych
    - Przydziały pamięci wykorzystujące stos
        - Powielanie łańcuchów znakowych na stosie
        - Tablice o zmiennej długości
    - Wybór mechanizmu przydzielania pamięci
    - Operacje na pamięci
        - Ustawianie wartości bajtów
        - Porównywanie bajtów
        - Przenoszenie bajtów
        - Wyszukiwanie bajtów
        - Manipulowanie bajtami
    - Blokowanie pamięci
        - Blokowanie fragmentu przestrzeni adresowej
        - Blokowanie całej przestrzeni adresowej
        - Odblokowywanie pamięci
        - Ograniczenia blokowania
        - Czy strona znajduje się w pamięci fizycznej?
    - Przydział oportunistyczny
        - Przekroczenie zakresu zatwierdzenia oraz stan braku pamięci (OOM)
- 10. Sygnały
    - Koncepcja sygnałów
        - Identyfikatory sygnałów
        - Sygnały wspierane przez system Linux
    - Podstawowe zarządzanie sygnałami
        - Oczekiwanie na dowolny sygnał
        - Przykłady
        - Uruchomienie i dziedziczenie
        - Odwzorowanie numerów sygnałów na łańcuchy znakowe

- Wysyłanie sygnału
  - Uprawnienia
  - Przykłady
  - Wysyłanie sygnału do samego siebie
  - Wysyłanie sygnału do całej grupy procesów
- Współużywalność
  - Funkcje, dla których współużywalność jest zagwarantowana
- Zbiory sygnałów
  - Inne funkcje obsługujące zbiory sygnałów
- Blokowanie sygnałów
  - Odzyskiwanie oczekujących sygnałów
  - Oczekiwanie na zbiór sygnałów
- Zaawansowane zarządzanie sygnałami
  - Struktura siginfo_t
  - Wspaniały świat pola si_code
- Wysyłanie sygnału z wykorzystaniem pola użytkowego
  - Przykład wykorzystania pola użytkowego
- Ułomność systemu Unix?

11. Czas
    - Struktury danych reprezentujące czas
      - Reprezentacja pierwotna
      - Następna wersja dokładność na poziomie mikrosekund
      - Kolejna, lepsza wersja dokładność na poziomie nanosekund
      - Wyłuskiwanie składników czasu
      - Typ danych dla czasu procesu
    - Zegary POSIX
      - Rozdzielczość źródła czasu
    - Pobieranie aktualnego czasu
      - Lepszy interfejs
      - Interfejs zaawansowany
      - Pobieranie czasu procesu
    - Ustawianie aktualnego czasu
      - Precyzyjne ustawianie czasu
      - Zaawansowany interfejs ustawiania czasu
    - Konwersje czasu
    - Dostrajanie zegara systemowego
    - Stan uśpienia i oczekiwania
      - Obsługa stanu uśpienia z dokładnością do mikrosekund
      - Obsługa stanu uśpienia z dokładnością do nanosekund
      - Zaawansowane zarządzanie stanem uśpienia
      - Przenośny sposób wprowadzania w stan uśpienia
      - Przepełnienia
      - Alternatywy stanu uśpienia
    - Liczniki
      - Proste alarmy
      - Liczniki interwałowe
      - Liczniki zaawansowane
        - Tworzenie licznika
        - Inicjalizowanie licznika
        - Odczytywanie czasu wygaśnięcia licznika

        - Odczytywanie wartości przepełnienia licznika
        - Usuwanie licznika
- A. Rozszerzenia kompilatora GCC dla języka C
    - GNU C
    - Funkcje wplatane (inline)
    - Zapobieganie wplataniu funkcji
    - Funkcje czyste (pure)
    - Funkcje stałe
    - Funkcje, które nie wracają do procedury wywołującej
    - Funkcje przydzielające pamięć
    - Wymuszanie sprawdzania wartości powrotnej dla procedur wywołujących
    - Oznaczanie funkcji niezalecanych
    - Oznaczanie funkcji używanych
    - Oznaczanie funkcji lub parametrów nieużywanych
    - Pakowanie struktury
    - Zwiększanie wartości wyrównania dla zmiennej
    - Umieszczanie zmiennych globalnych w rejestrach
    - Optymalizacja gałęzi kodu
    - Uzyskiwanie typu dla wyrażenia
    - Uzyskiwanie wielkości wyrównania dla danego typu
    - Pozycja elementu w strukturze
    - Uzyskiwanie powrotnego adresu funkcji
    - Zakresy funkcji case
    - Arytmetyka wskaźników do funkcji oraz wskaźników void
    - Więcej przenośności i elegancji za jednym razem
- B. Bibliografia
    - Programowanie w języku C
    - Programowanie w Linuksie
    - Jądro Linuksa
    - Projektowanie systemu operacyjnego
- C. O autorze
- Indeks
- Kolofon
- Copyright