

# Linux - wprowadzenie do systemu operacyjnego

---

## 1. Wielodostępne systemy operacyjne

Nowoczesne systemy operacyjne są w większości systemami wielodostępnymi, które pozwalają pracować jednocześnie wielu użytkownikom za pośrednictwem terminali podłączonych bezpośrednio do komputera lub poprzez sieć komputerową.

Artykuł ten stanowi wprowadzenie do problematyki, którą przedstawimy w dalszych artykułach. Przedstawiamy tu ogólną charakterystykę wielodostępnych systemów operacyjnych. Uzasadnimy też wybór systemu Linux jako reprezentanta tej klasy systemów w naszych rozważaniach. Na koniec charakteryzujemy sesję użytkownika w systemie Linux, opisując sposób autoryzacji użytkowników, atrybuty użytkownika oraz interfejs tekstowy i graficzny systemu.

Systemy operacyjne można podzielić najogólniej na dwie kategorie:

- systemy specjalnego przeznaczenia,
- systemy ogólnego przeznaczenia.

Pierwsza kategoria obejmuje systemy opracowane specjalnie do wykonywania jakiegoś konkretnego zadania np.:

- systemy sterowania procesami przemysłowymi czy złożonymi urządzeniami w czasie rzeczywistym,
- systemy przetwarzania transakcji.

Takie systemy są wyspecjalizowanymi programami i często nie są nawet uważane za systemy operacyjne pomimo, że bezpośrednio sprawują pełną kontrolę nad sprzętem.

Systemy ogólnego przeznaczenia pozwalają wykorzystywać system komputerowy do różnych celów w zależności od potrzeb użytkowników

Można wśród nich wyróżnić:

- systemy dla indywidualnego użytkownika,
- systemy wsadowe,
- systemy wielodostępne.

Nowoczesne systemy operacyjne są w większości **systemami wielodostępnymi**, które pozwalają pracować jednocześnie wielu użytkownikom za pośrednictwem terminali podłączonych bezpośrednio do komputera lub poprzez sieć komputerową. Cechę tę posiadają takie systemy, jak: Unix, Linux, Windows 2000, MacOS X i inne.

Podstawowe informacje dotyczące systemów operacyjnych zostały zamieszczone w podręczniku Informatyka 1 [1]. Niniejszy podręcznik poświęcamy omówieniu w sposób bardziej szczegółowy złożonych zagadnień występujących w systemach wielodostępnych. Zagadnienia te przedstawimy posługując się przykładem systemu Linux.

### (1.1) Cele i zadania

Każdy system operacyjny ma do spełnienia dwa podstawowe cele:

- zapewnienie wygodnej pracy użytkownikom,
- efektywne zarządzanie zasobami systemu komputerowego.

Cele te pozostają niestety w sprzeczności ze sobą. Im bardziej „przyjazny” staje się system operacyjny oferując użytkownikom wciąż więcej udogodnień, tym więcej zasobów sam zużywa zamiast udostępniać je użytkownikom. Dotyczy to w szczególności zasobów pamięci operacyjnej i dyskowej oraz wykorzystania procesora. Osiągnięcie właściwego kompromisu jest szczególnie ważne w systemach wielodostępnych, gdy wielu użytkowników korzysta z ograniczonych zasobów systemu.

Osiągnięcie powyższych celów wiąże się z realizacją określonych zadań obejmujących:

- zarządzanie zadaniami,
- zarządzanie pamięcią operacyjną i pomocniczą,
- zarządzanie systemem wejścia-wyjścia,
- zarządzanie plikami,
- pracę sieciową,
- ochronę zasobów,
- komunikację z użytkownikami.

## **(1.2) Interfejsy programowe**

System operacyjny udostępnia swoje usługi użytkownikom i ich programom poprzez interfejsy programowe.

**Funkcje systemowe** stanowią interfejs pomiędzy programami a jądrem systemu operacyjnego. Umożliwiają programom korzystanie z usług jądra i sprzętu komputerowego bez naruszania bezpieczeństwa systemu.

Programy systemowe tworzą z kolei interfejs dla użytkowników. W jego skład wchodzi między innymi tekstowe interpretry poleceń oraz programy tworzące interfejs graficzny systemu. Programy systemowe umożliwiają użytkownikom wykonywanie typowych operacji dotyczących manipulowania plikami, przetwarzania ich zawartości, tworzenia i wykonywania programów, komunikacji czy informowania o stanie systemu.

## **2. Charakterystyka systemu Linux**

System Linux zdobywa coraz większe uznanie dużej grupy użytkowników. O jego sukcesie decyduje wiele czynników a między innymi:

- efektywność i stabilność systemu,
- powszechna dostępność bez jakichkolwiek opłat licencyjnych,
- bogaty zestaw oprogramowania umożliwiający szeroki zakres zastosowań,
- możliwość pracy na wielu platformach sprzętowych przy stosunkowo niewielkich wymaganiach,
- możliwość łatwej współpracy z innymi popularnymi systemami operacyjnymi,
- bogata dokumentacja w wersji elektronicznej,
- dostępność kodu źródłowego.

### **(2.1) Historia systemu Linux**

System Linux jest dziełem wielu programistów z całego świata. Podstawy systemu stworzył Linus Torvalds, student uniwersytetu w Helsinkach, który w 1991 roku napisał pierwszą wersję jądra systemu. Wzorował się przy tym na akademickim systemie Minix i zachował z nim znaczną zgodność. Linux został udostępniony w Internecie na zasadach licencji GPL (ang. General Public License). System wzbudził duże zainteresowanie programistów, co zapoczątkowało jego burzliwy rozwój.

Projekt systemu Linux jest dość ściśle związany z projektem GNU, który zakładał opracowanie darmowego systemu operacyjnego wzorowanego na Unixie. W ramach projektu powstało wiele pakietów oprogramowania, które są obecnie dostępne w systemie Linux.

Twórcy Linuxa nie korzystali bezpośrednio z kodu systemu Unix. Linux został jednak zaprojektowany zgodnie ze standardem POSIX 1003.1, który definiuje interfejsy programowania i narzędzi systemowych dla systemu Unix.

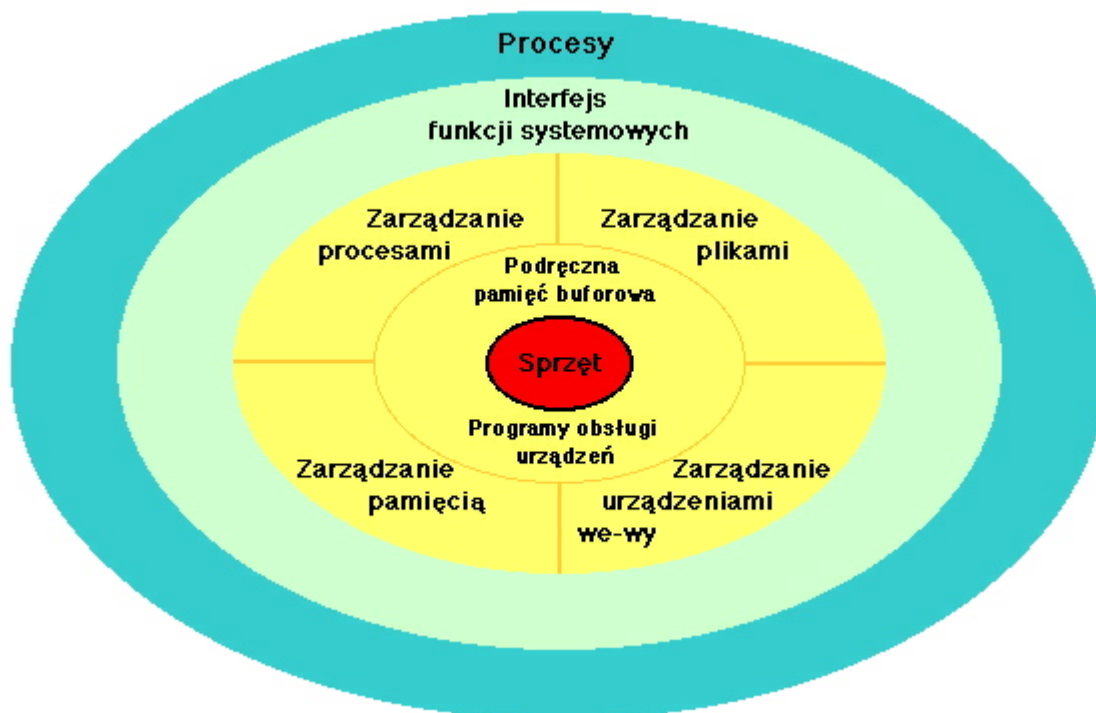
### **(2.2) Najważniejsze cechy systemu**

Linux jest systemem nowoczesnym, ponieważ wielu programistów wciąż pracuje nad rozwojem jądra systemu oraz bogatego zestawu programów usługowych. O jego nowoczesności decydują przede wszystkim cechy wymienione poniżej:

- wielodostęp,
- wielozadaniowość, czyli praca z podziałem czasu procesora pomiędzy wiele zadań,
- wieloprzetwarzanie, czyli praca wieloprocessorowa,
- możliwość uruchamiania zadań w łagodnym czasie rzeczywistym,
- obsługa różnych typów systemów plików,
- obsługa różnych protokołów sieciowych,
- obsługa różnych formatów plików wykonywalnych,
- wykorzystanie współdzielonych bibliotek.

### (2.3) Struktura systemu

Systemy Unix i Linux mają strukturę warstwową przedstawioną na rys. 1.1.



Rys. 1.1 Warstwowa struktura systemów Unix

Warstwa programów obsługi urządzeń komunikuje się bezpośrednio ze sprzętem komputerowym, więc musi uwzględniać specyficzne cechy tego sprzętu. Programy obsługi muszą być napisane w języku programowania niskiego poziomu oddzielnie dla każdej platformy sprzętowej. Cała reszta systemu została zaimplementowana w języku C.

Jądra systemów Unix i Linux są monolityczne. Chociaż można w nich wyróżnić funkcjonalne części składowe, to stanowią one jeden program i korzystają ze wspólnych struktur danych. Takie rozwiązanie zapewnia największą efektywność systemu. Do podstawowych zadań jądra należy:

- zarządzanie procesami,
- zarządzanie pamięcią,
- zarządzanie plikami,
- zarządzanie urządzeniami wejścia-wyjścia.

Kod jądra Linuxa może być podzielony fizycznie na kilka części:

- jądro podstawowe,
- moduły jądra, dołączane dynamicznie.

Modularność jądra pozwala na elastyczne dostosowywanie jego funkcji do zmieniających się wymagań bez konieczności ponownej kompilacji całego kodu. Moduły mogą zawierać programy obsługi nowych urządzeń podłączonych do systemu lub interfejsy programowe różnych typów systemów plików.

### 3. Sesja użytkownika w systemie Linux

#### (3.1) Autoryzacja użytkowników

System Linux przeznaczony jest dla wielu użytkowników, którzy mogą w dodatku pracować jednocześnie. Pociąga to za sobą konieczność autoryzacji dostępu do systemu i ochrony zasobów każdego użytkownika. Użytkownik musi być zarejestrowany w systemie i posiadać własne konto, zabezpieczone hasłem dostępu. Autoryzacja oparta jest na sprawdzaniu nazw i haseł.

System poprosi najpierw o wpisanie starego hasła, a następnie o dwukrotne wprowadzenie nowego hasła. Zwykły użytkownik może oczywiście zmienić wyłącznie hasło do swojego konta. Administrator może wywołać polecenie **passwd** z nazwą dowolnego użytkownika, aby zmienić jego hasło.

System Linux po uruchomieniu nie jest bezpośrednio dostępny dla użytkowników, tylko oczekuje na ich zgłoszenia. Zgłaszając się do systemu, użytkownik podaje swoją nazwę i hasło. System dokonuje autoryzacji poprzez:

- sprawdzenie, czy użytkownik o podanej nazwie jest zarejestrowany w systemie w pliku **/etc/passwd**,
- zakodowanie podanego hasła,
- porównanie z zakodowanym hasłem przechowywanym przez system w pliku **/etc/shadow**.

W wyniku pomyślnej autoryzacji system uruchamia sesję użytkownika, udostępniając jego konto. Proces zgłaszania i autoryzacji użytkownika określa się jako **logowanie** do systemu.

W każdym systemie istnieje konto uprzywilejowanego użytkownika o nazwie **root** i identyfikatorze UID = 0, określanego jako nadzorca (ang. supervisor), superużytkownik (ang. superuser) lub zwyczajowo root. Konto to jest przeznaczone dla administratora systemu, który dysponuje nieograniczonymi uprawnieniami. Jednym z jego zadań jest zakładanie kont nowym użytkownikom. Może w tym celu wykorzystać polecenie systemowe **useradd**.

#### **useradd [opcje] użytkownik**

Atrybuty nowego konta podaje się poprzez opcje wywołania lub odpowiadając na pytania działającego programu.

Użytkownik może zmienić swoje hasło posługując się poleceniem **passwd**.

#### **passwd [użytkownik]**

System poprosi najpierw o wpisanie starego hasła, a następnie o dwukrotne wprowadzenie nowego hasła. Zwykły użytkownik może oczywiście zmienić wyłącznie hasło do swojego konta. Administrator może wywołać polecenie **passwd** z nazwą dowolnego użytkownika, aby zmienić jego hasło.

#### (3.2) Atrybuty użytkownika

System przechowuje w pliku **/etc/passwd** następujące atrybuty każdego zarejestrowanego użytkownika:

<b>nazwa</b>	-nazwa jednoznacznie identyfikująca konto użytkownika,
<b>identyfikator użytkownika</b>	-numer jednoznacznie identyfikujący użytkownika w systemie,
<b>UID</b>	
<b>identyfikator grupy</b>	-numer grupy, do której należy użytkownik,
<b>GID</b>	
<b>katalog domowy</b>	-prywatny katalog użytkownika, w którym może bezpiecznie przechowywać swoje pliki, zabezpieczone przed dostępem innych użytkowników,

## powłoka logowania

-nazwa interpretera poleceń, który jest uruchamiany po zalogowaniu użytkownika.

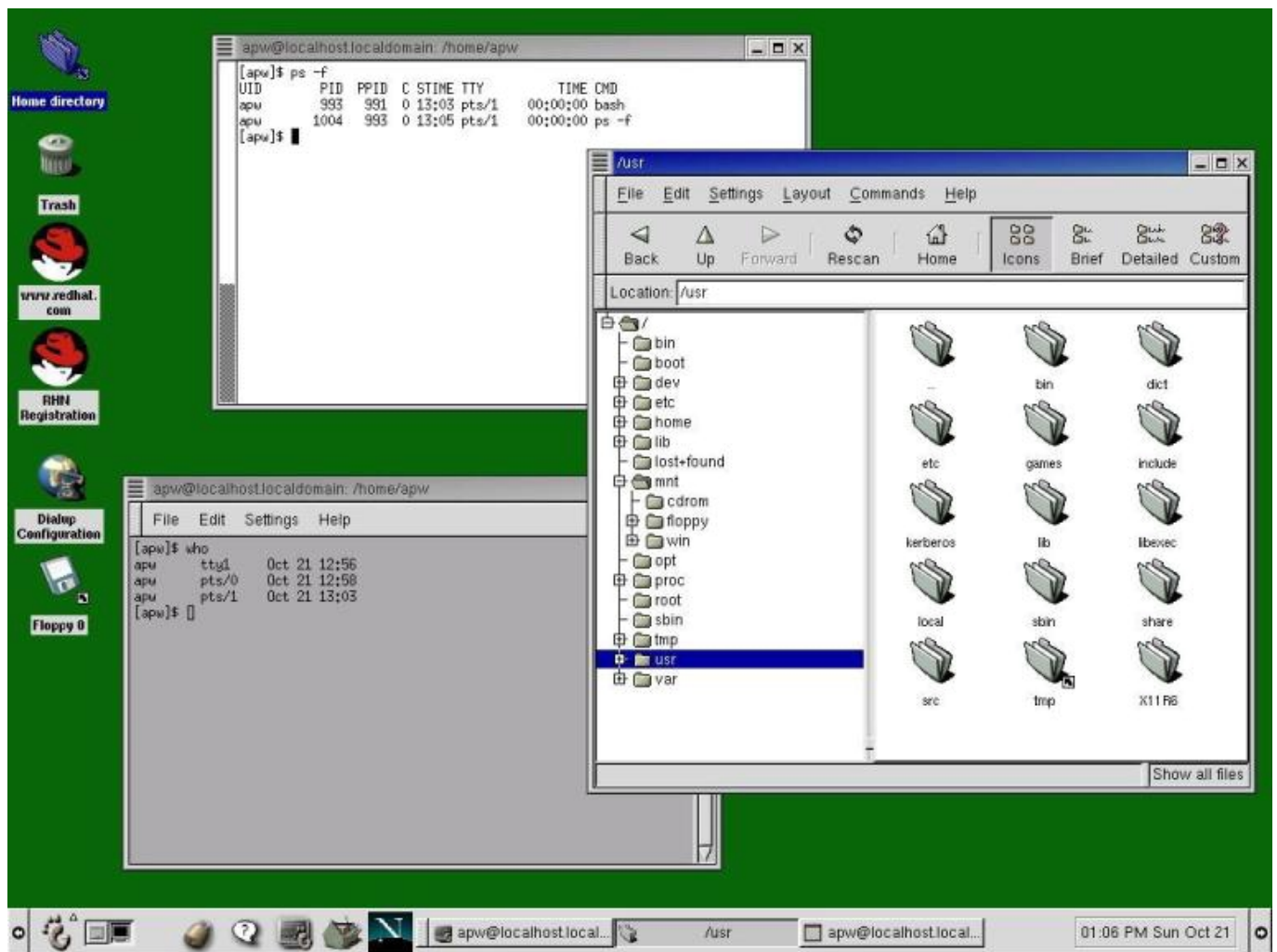
System dzieli użytkowników na grupy, aby zwiększyć w ten sposób możliwość różnicowania uprawnień do uruchamiania programów i korzystania z plików. Informacje o członkach poszczególnych grup przechowywane są w pliku **/etc/group**.

### (3.3) Środowisko graficzne X Window

Standardowym środowiskiem graficznym systemu Unix jest system X Window. Jego aktualna wersja określana jest nazwą X11 Release 6. Linux dysponuje darmową wersją tego środowiska o nazwie XFree86, która zachowuje pełną zgodność z wersją X11.

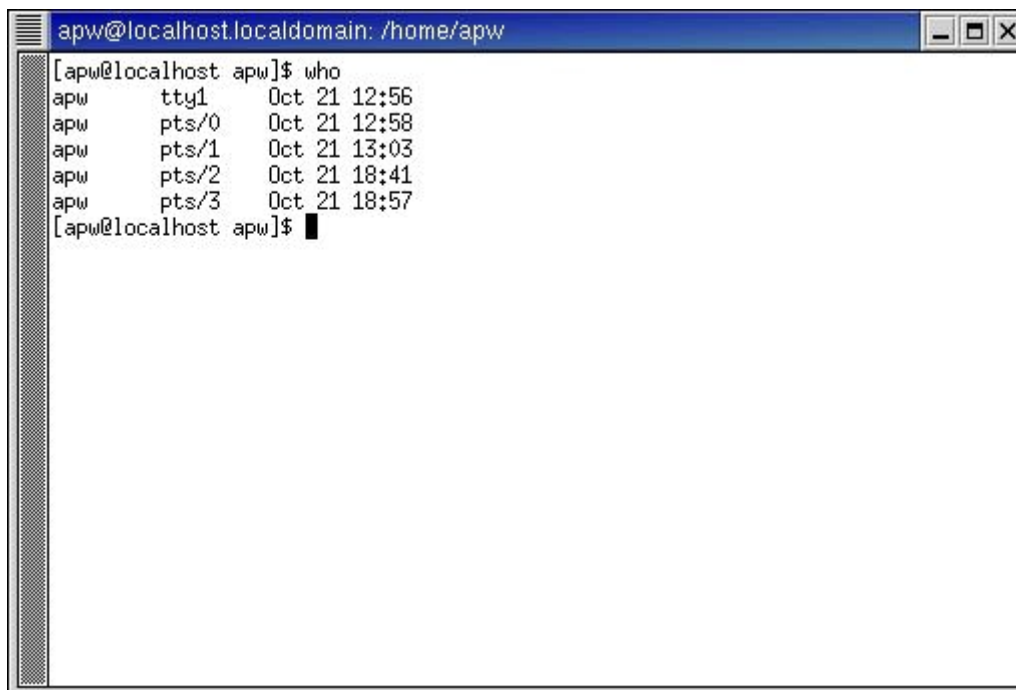
Po zalogowaniu do systemu użytkownik pracujący przy konsoli komputera może uruchomić środowisko graficzne poleceniem **STARTx**. System operacyjny uruchamia wtedy serwer systemu X Windows o nazwie X dostosowany do parametrów karty graficznej komputera.

Uruchamianiem i zarządzaniem oknami w środowisku graficznym zajmuje się program menedżera okien. Obecnie coraz większą popularność zdobywają systemy aktywnego pulpitu takie, jak KDE i GNOME. Rys. 1.2 prezentuje przykładowy wygląd pulpitu GNOME.



Rys. 1.2 System aktywnego pulpitu GNOME

Jednym z najważniejszych programów jest program emulacji terminalu, który pozwala otworzyć okno z uruchomionym interpreterem poleceń. Użytkownik uzyskuje w ten sposób możliwość uruchamiania programów, które nie korzystają wprost z okienkowego interfejsu graficznego. Najpopularniejszym emulatorem terminalu jest obecnie program **xterm**, który stał się praktycznie standardem we wszystkich wersjach systemu Unix i wszystkich dystrybucjach Linuxa. Na rys. 1.3 prezentujemy przykładowe okno programu **xterm** z uruchomionym interpreterem **bash**.

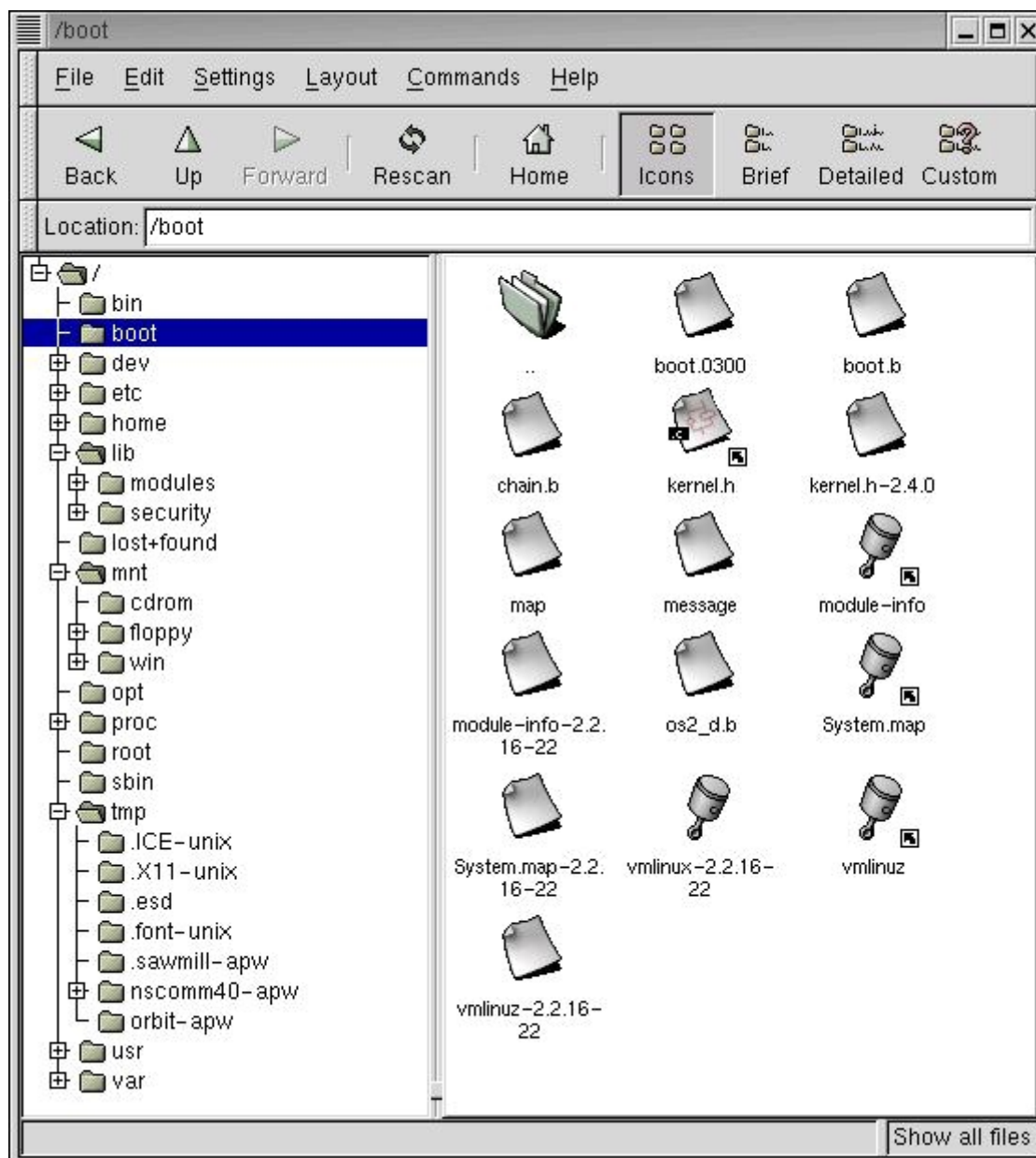


```
apw@localhost.localdomain: /home/apw
[apw@localhost apw]$ who
apw    tty1    Oct 21 12:56
apw    pts/0    Oct 21 12:58
apw    pts/1    Oct 21 13:03
apw    pts/2    Oct 21 18:41
apw    pts/3    Oct 21 18:57
[apw@localhost apw]$ █
```

Rys. 1.3 Okno emulatora terminalu **xterm**

Systemy oferują na ogół również inne emulatory terminalu specyficzne dla konkretnej implementacji systemu X Window. Na przykład system aktywnego pulpitu GNOME zawiera własny emulator o nazwie GNOME Terminal, który można uruchomić z poziomu menu lub poleceniem **gnome-terminal**.

Innym przydatnym programem jest graficzny menedżer plików, który umożliwia oglądanie zawartości katalogów, manipulację plikami, przeglądanie i edycję zawartości plików. Jednym z takich programów jest GNU Midnight Commander pokazany na rys. 1.4.



Rys. 1.4 Program menedżera plików GNU Midnight Commander

### (3.4) Dokumentacja w formie elektronicznej

Dokumentacja standardowych programów usługowych, funkcji systemowych i dodatkowego oprogramowania dostępna jest w postaci odpowiednio sformatowanych plików dla programów **man** oraz **info**. Dostępne są zbiory tekstów znane pod ogólną nazwą HOWTO opisujące rozwiązania najczęściej spotykanych problemów.

## 4. Bibliografia

### (4.1) Bibliografia podstawowa

1. Dąbrowski W., Nowak S., Michalak G.: Informatyka 1, Ośrodek Kształcenia Na Odległość, Politechnika Warszawska 2001,

2. Silberschatz A., Galvin P.B.: Podstawy systemów operacyjnych, WNT 2000 (rozdziały: 1, 3, 22),
3. Petersen R.: Arkana Linux, Wydawnictwo RM 1997 (rozdział: 1, 3, 4).

#### (4.2) Bibliografia uzupełniająca:

1. Główny serwer projektu Linux : [www.linux.org](http://www.linux.org)
2. Grupy dyskusyjne (USENET) :
  - o [pl.comp.os.linux](http://pl.comp.os.linux)
  - o [pl.comp.os.linux.programowanie](http://pl.comp.os.linux.programowanie)
  - o [pl.comp.os.linux.sieci](http://pl.comp.os.linux.sieci)
  - o [pl.comp.os.linux.sprzet](http://pl.comp.os.linux.sprzet)
3. Serwery polskich Grup Użytkowników Linux-a (Linux User Groups)
  - o [www.linux.org.pl](http://www.linux.org.pl)
  - o [www.linux.bielsko.pl](http://www.linux.bielsko.pl)
  - o [www.linux.bydgoszcz.org](http://www.linux.bydgoszcz.org)
  - o [www.krakow.linux.org.pl](http://www.krakow.linux.org.pl)
  - o [www.silesia.linux.org.pl](http://www.silesia.linux.org.pl)
  - o [www.torun.linux.org.pl](http://www.torun.linux.org.pl)
  - o [www.linux.gda.pl](http://www.linux.gda.pl)
  - o [www.wroclaw.linux.org.pl](http://www.wroclaw.linux.org.pl)

---

Autorami powyższego artykułu są mgr inż. A. Wielgus i dr Z. Jaworski. Drobnych zmian i poprawek dokonał Ł. Fronczyk  
W dziale źródła/C++ znajdują się przykładowe programy do tego i pozostałych artykułów z tego cyklu.  
Wszelkie pytania proszę kierować na adres [L.Fronczyk@elka.pw.edu.pl](mailto:L.Fronczyk@elka.pw.edu.pl)

Źródło: [4programmers.net](http://4programmers.net). Treść udostępniona na zasadach licencji [Creative Commons Attribution](http://creativecommons.org/licenses/by/3.0/)

---

Autor: A. Wielgus i dr Z. Jaworski  
Artykuł pobrano ze strony [eioba.pl](http://eioba.pl)