

2. Przegląd współczesnych systemów operacyjnych

Wstęp

Przedstawiamy porównanie cech najpopularniejszych współczesnych systemów operacyjnych: UNIX, Linux, Windows i MacOS. Opisujemy historię systemu Linux i jego najważniejsze cechy. Staramy się jednocześnie uzasadnić wybór tego systemu jako reprezentanta współczesnych systemów operacyjnych w dalszych rozważaniach. Na koniec charakteryzujemy sesję użytkownika w systemie Linux, opisując sposób autoryzacji użytkowników, atrybuty użytkownika oraz interfejs tekstowy i graficzny systemu.

2.1. System UNIX

System UNIX jest jednym z najstarszych, wykorzystywanych współcześnie systemów operacyjnych. Pierwsza wersja systemu, przeznaczona dla komputera PDP 7, została opracowana w 1969 r. przez Kena Thompsona i Dennisa Ritchie z firmy Bell Laboratories, należącej do AT&T. W 1973 r. kod systemu UNIX został przepisany w nowym języku programowania C, opracowanym specjalnie w tym celu. Początkowo UNIX był rozwijany wyłącznie na potrzeby firmy Bell Laboratories i zyskiwał w niej coraz większą popularność. Został też przeniesiony na nowe komputery np. PDP 11 i Interdata. Kolejne wersje systemu nosiły oznaczenia UNIX Edycja 1, Edycja 2 ... (lub UNIX Wersja 1, ...). Dopiero wersje UNIX Edycja 6 i Edycja 7 zostały udostępnione poza firmą i zyskały dużą popularność, zwłaszcza na uniwersytetach w USA i Australii. Na szczególną uwagę zasługiwał UNIX Edycja 7 z 1978 r. opracowany już dla komputerów 32-bitowych. Niektóre uniwersytety i firmy informatyczne podjęły nawet dalsze prace nad rozwojem UNIXa lub własnych, wzorowanych na nim systemów. Najbardziej zaawansowane okazały się prace w University of California w Berkeley. Dalszy rozwój systemu potoczył się zasadniczo dwoma drogami: w AT&T i na uniwersytecie w Berkeley.

W AT&T opracowano w 1982 r. nową wersję systemu o nazwie UNIX **System III**, przeznaczoną do dystrybucji poza firmą. W 1983 r. AT&T wydało UNIX **System V**. Nazwa ta została już zachowana w kolejnych wydaniach, uzupełniana jedynie przez numer kolejnej wersji: System V Wersja 2 (w skrócie SVR2 lub SV.2) w 1984, System V Wersja 3 (SVR3) w 1987, System V Wersja 4 (SVR4) w 1989 i wreszcie ostatnia wersja System V Wersja 4.2/ES/MP (SVR4.2/ES/MP) w 1993 r.

Uniwersytet w Berkeley rozwijał własną odmianę systemu UNIX rozprowadzaną pod nazwą **BSD** (Berkeley Software Distribution). Pierwsza wersja 1BSD, oparta na Edycji 6 systemu UNIX, pojawiła się już w 1978 roku. Znacznie rozszerzona wersja 3BSD z 1979 r. została oparta na Edycji 7 i przeznaczona dla komputerów VAX. W 1980 wydana została wersja 4BSD, stanowiąca podstawę wszystkich późniejszych opracowań systemu UNIX z Berkeley. Ostatnie wydanie z 1993 r. nosi oznaczenie 4.4BSD. Uniwersytet w Berkeley wniósł ogromny wkład w rozwój systemu UNIX. W systemach BSD wprowadzono po raz pierwszy do UNIX-a wiele nowych mechanizmów m.in. pamięć wirtualną, stronicowanie na żądanie, interfejs gniazd komunikacji sieciowej.

Wielokrotnie podejmowano próby standaryzacji interfejsów systemu UNIX. Powstało w efekcie kilka konkurujących ze sobą standardów, które przedstawiono w Tabl. 2.1.

Tablica 2.1 Standardy systemu UNIX

Standard	Opis
SVID (System V Interface Definition)	Standard opracowany w 1985 r. przez AT&T. Opisuje działanie oryginalnego systemu UNIX System V w zakresie funkcji systemowych, funkcji z biblioteki C, programów systemowych i urządzeń. Ostatnia wersja SVID4 (1995) była zgodna z innymi standardami XPG4 i POSIX 1003.1-1990.
4.x BSD (Berkeley Software Distribution)	Standard opisujący system 4 BSD z Uniwersytetu z Berkeley.
XPG (X/Open Portability Guide)	Standard opracowany w 1985 r. przez organizację X/OPEN skupiającą producentów komputerów z systemem UNIX. Wersja XPG4 (1992) stała się później podstawą standardu SUS.
POSIX (Portable Operating System Interface)	Standard opracowany w 1988 r. przez IEEE i ISO jako IEEE Std. 1003.1 (określany również jako POSIX 1003.1 lub POSIX.1). Opisuje interfejsy programowania i narzędzi systemowych dla systemu UNIX. Ostatnia wersja to POSIX.1-2008.

SUS
(Single UNIX
Specification)

Standard opracowany w 1994 przez Open Group na bazie XPG4. Począwszy od wersji SUSv3 rozwijany wspólnie z POSIX.1 przez Austin Group. Ostatnia wersja SUSv4 pokrywa standard POSIX.1-2008.

W miarę najpełniejszy wydaje się standard SUS, który w najnowszej wersji SUS4 obejmuje w większości również pozostałe standardy.

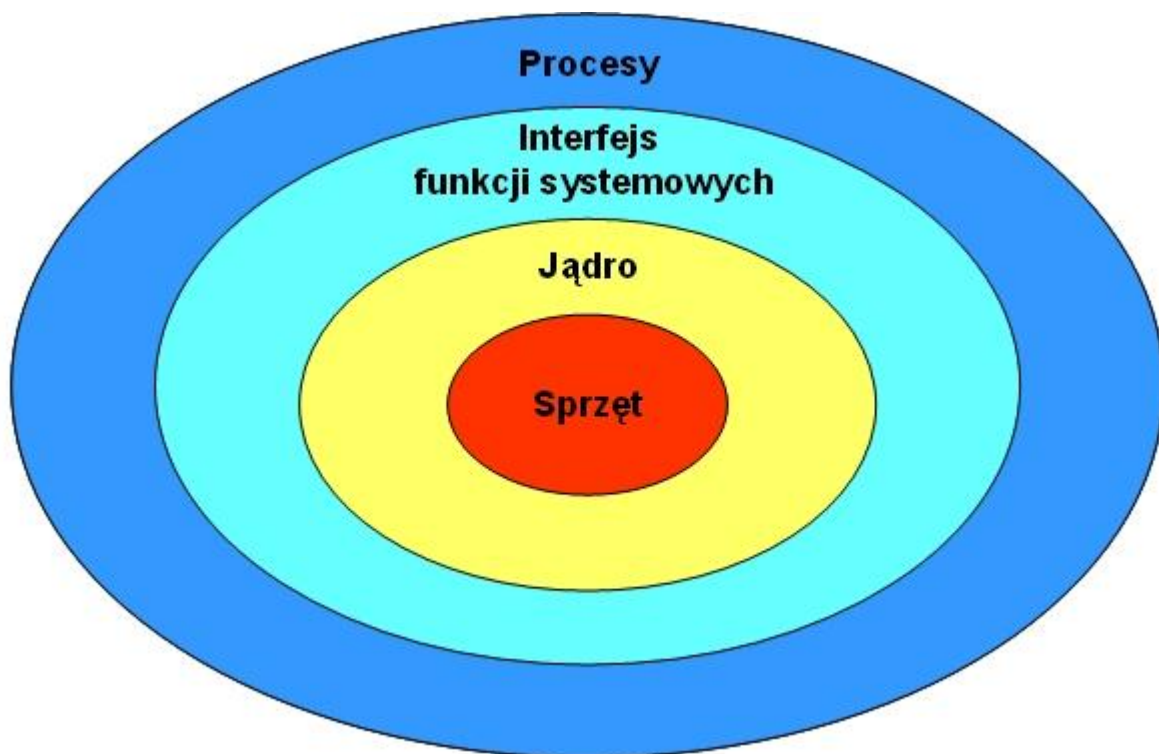
Oprócz dwóch klasycznych wersji systemu UNIX, System V Wersja 4 i 4.4 BSD, dostępnych jest obecnie wiele komercyjnych systemów uniksowych opracowanych na bazie różnych wersji Systemu V lub 4.x BSD. Większość dużych firm komputerowych oferuje własną wersję takiego systemu np.:

- Sun OS i Solaris firmy Sun,
- AIX firmy IBM,
- ULTRIX, OSF/1 i Digital UNIX firmy DEC,
- HPUNIX firmy HP,
- A/UX i Mac OS X firmy Apple,
- SCO UNIX i UnixWare firmy SCO,
- UNICOS firmy Cray (producent superkomputerów Cray).

Pomimo swojej ponad czterdziestoletniej historii, system UNIX wciąż pozostaje nowoczesnym i w pełni funkcjonalnym systemem operacyjnym, przeznaczonym dla wszystkich klas komputerów. Jest to możliwe dzięki stałemu rozwojowi systemu wspieranemu przez wiele instytucji. O jego nowoczesności decydują przede wszystkim cechy wymienione poniżej:

- wielodostęp,
- wielozadaniowość,
- wieloprzetwarzanie symetryczne,
- możliwość uruchamiania zadań w łagodnym czasie rzeczywistym,
- obsługa różnych typów systemów plików,
- obsługa różnych protokołów sieciowych,
- obsługa różnych formatów plików wykonywalnych.

System UNIX ma strukturę warstwową przedstawioną na Rys. 2.1.



Rys. 2.1 Warstwowa struktura systemu UNIX

Zewnętrzną warstwę systemu stanowią wszystkie programy systemowe i użytkowe uruchamiane przez użytkowników jako procesy. W tej warstwie zrealizowane są interpretry poleceń (powłoki) oraz graficzny interfejs użytkownika (X Window). Warstwa funkcji systemowych stanowi interfejs pomiędzy programami a jądrem systemu operacyjnego, umożliwiając im korzystanie z usług jądra i sprzętu komputerowego bez naruszania bezpieczeństwa systemu. Do podstawowych zadań jądra należy:

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

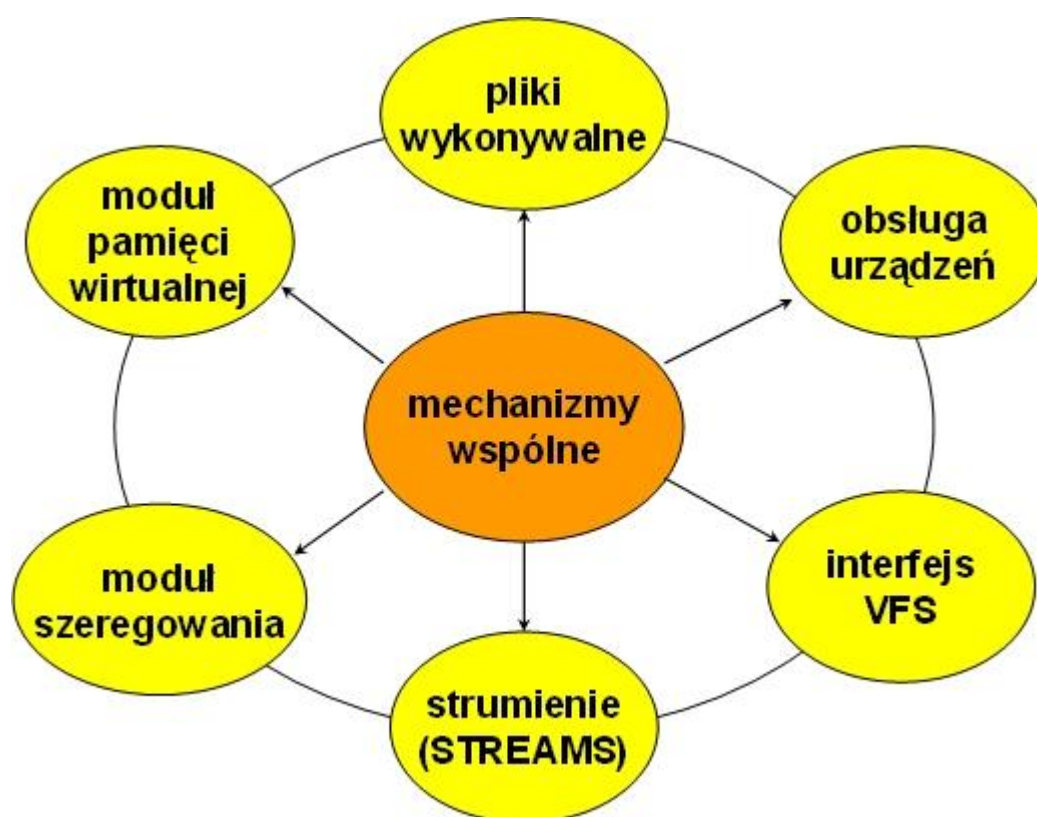
W tradycyjnych wersjach systemu UNIX jądro było monolityczne (Rys.2.2). Choć można w nich było wyróżnić funkcjonalne części składowe, to stanowiły one jeden program i korzystały ze wspólnych struktur danych. Takie rozwiązanie zapewniało największą efektywność systemu.

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.

Współczesne wersje systemu UNIX mają jądro o budowie modularnej, przedstawione na Rys. 2.3.



Rys. 2.2 Tradycyjne jądro monolityczne systemu UNIX



Rys. 2.3 Nowoczesne jądro modułarne systemu UNIX

2.2. System MS Windows

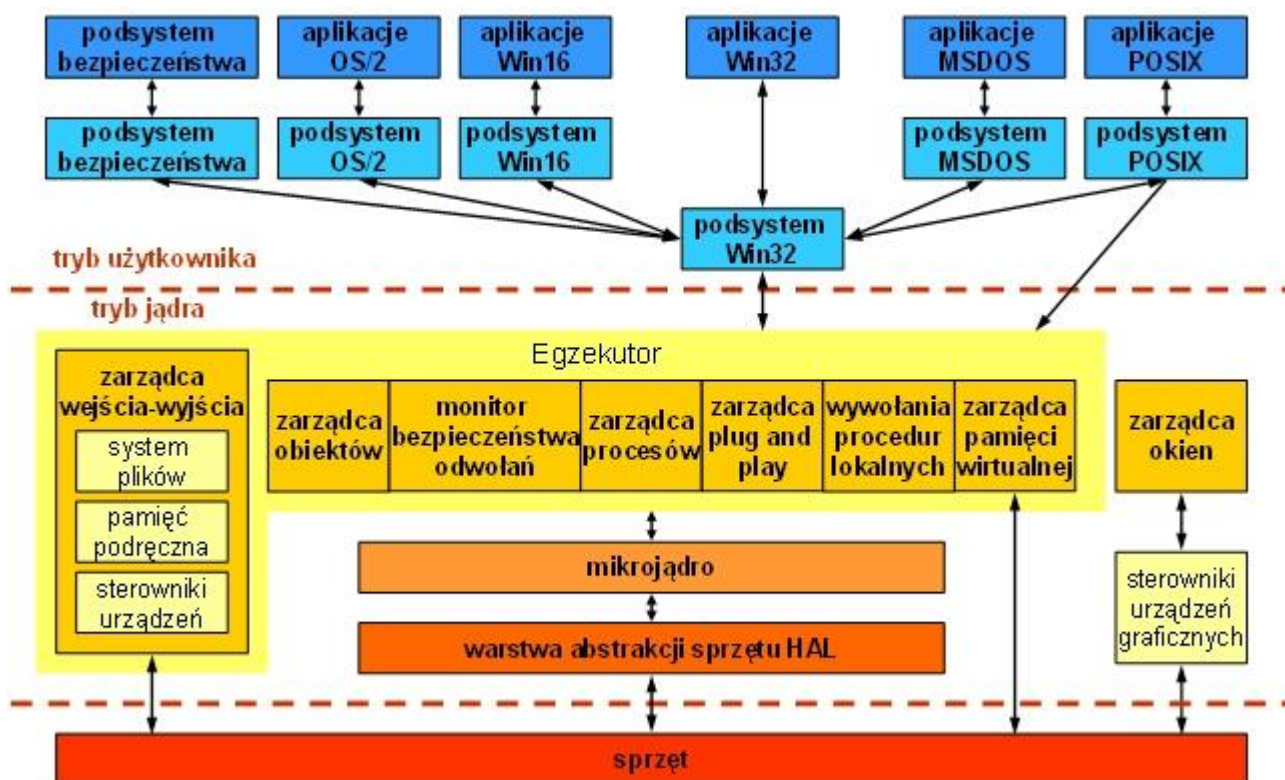
Pierwsza wersja systemu Windows firmy Microsoft pojawiła się w 1985 r. Podstawę stanowił popularny system operacyjny komputerów osobistych MS-DOS, do którego dodano graficzny interfejs użytkownika. W kolejnych wersjach systemu zachowywano daleko idącą zgodność z systemem MS-DOS, co stanowiło istotne ograniczenie rozwojowe. Pomimo wprowadzania wielu zmian w projekcie systemu, wciąż nie zapewniał on wielozadaniowości ani właściwej ochrony pamięci. Miało to zdecydowanie niekorzystny wpływ na stabilność i niezawodność systemu. Ta linia rozwojowa była kontynuowana w postaci kolejnych wersji: Windows 3.x, 95, 98, Me.

Dopiero w 1993 r. firma Microsoft wprowadziła zupełnie nowy system operacyjny o nazwie Windows NT (wersja 3.1), który zapoczątkował nową, rozwijaną niezależnie, linię systemów. Był to system wielozadaniowy z 32-bitowym adresowaniem oraz ochroną pamięci. Zapewniał dużą niezawodność, rozszerzalność i przenośność oraz zgodność programową z innymi systemami operacyjnymi (DOS, Windows, OS/2, POSIX). System NT został przeznaczony przede wszystkim dla stacji roboczych i serwerów. Dużą popularność zdobyła szczególnie wersja NT 4.0, w której wprowadzono ulepszony interfejs graficzny z systemu Windows 95. Kontynuację tej linii rozwojowej stanowią kolejne wersje systemu, które zostały wprowadzone do sprzedaży pod nowymi nazwami:

- Windows 2000 (Windows NT 5.0),
- Windows XP (Windows NT 5.1),
- Windows Server 2003 (Windows NT 5.2),
- Windows XP Professional x64 (Windows NT 5.2),
- Windows Vista (Windows NT 6.0),
- Windows Server 2008 (Windows NT 6.0),
- Windows 7 (Windows NT 6.1),
- Windows Server 2008 R2 (Windows NT 6.1),
- Windows 8 (Windows NT 6.2),
- Windows 8.1 (Windows NT 6.3),
- Windows 10 (Windows NT 6.4).

Obecnie rodzina systemów Windows reprezentowana jest już tylko przez jedną linię rozwojową. Systemy Windows 7, 8 i 10 to uniwersalne systemy wielozadaniowe przeznaczone dla stacji roboczych, serwerów oraz komputerów osobistych.

System Windows NT miał strukturę modułową. Projekt pierwszej wersji zakładał podział systemu na wiele równorzędnych modułów komunikujących się ze sobą za pomocą komunikatów poprzez wyróżniony moduł centralny, zwany mikrojądrem. Jedynie mikrojądro działało w trybie chronionym, a wszystkie pozostałe moduły pracowały w trybie użytkownika, pełniąc na przemian rolę klienta lub serwera. Strukturę taką określa się jako strukturę klient-serwer lub strukturę opartą na mikrojądrze. Wydajność takiego systemu okazała się jednak niewystarczająca. W nowszych wersjach zmodyfikowano strukturę systemu, przemieszczając część modułów do trybu jądra i rezygnując z przesyłania komunikatów między nimi na rzecz wspólnej pamięci. Zmodyfikowaną strukturę przedstawiono na rysunku 2.4.



Rys. 2.4 Struktura systemu Windows 7

W trybie jądra działają:

- warstwa abstrakcji sprzętu - ukrywa cechy sprzętu i tworzy interfejs maszyny wirtualnej dla innych modułów systemu,
- mikrojądro - zajmuje się planowaniem procesów i wątków oraz ich synchronizacją, obsługuje przerwania i sytuacje wyjątkowe
- egzekutor - dostarcza usług systemowych w zakresie zarządzania procesami, pamięcią wirtualną, obiektami i systemem wejścia-wyjścia oraz ułatwienia wywołań procedur lokalnych.

W trybie użytkownika działa scentralizowany podsystem bezpieczeństwa oraz podsystemy środowiskowe, implementujące interfejsy programu użytkowego API różnych systemów operacyjnych:

- Win32 - podstawowe środowisko systemu Windows NT,
- Win16 - środowisko 16-bitowych systemów Windows,
- DOS - środowisko systemu DOS,
- OS/2 - środowisko systemu OS/2,
- POSIX - środowisko zgodne ze standardem POSIX.

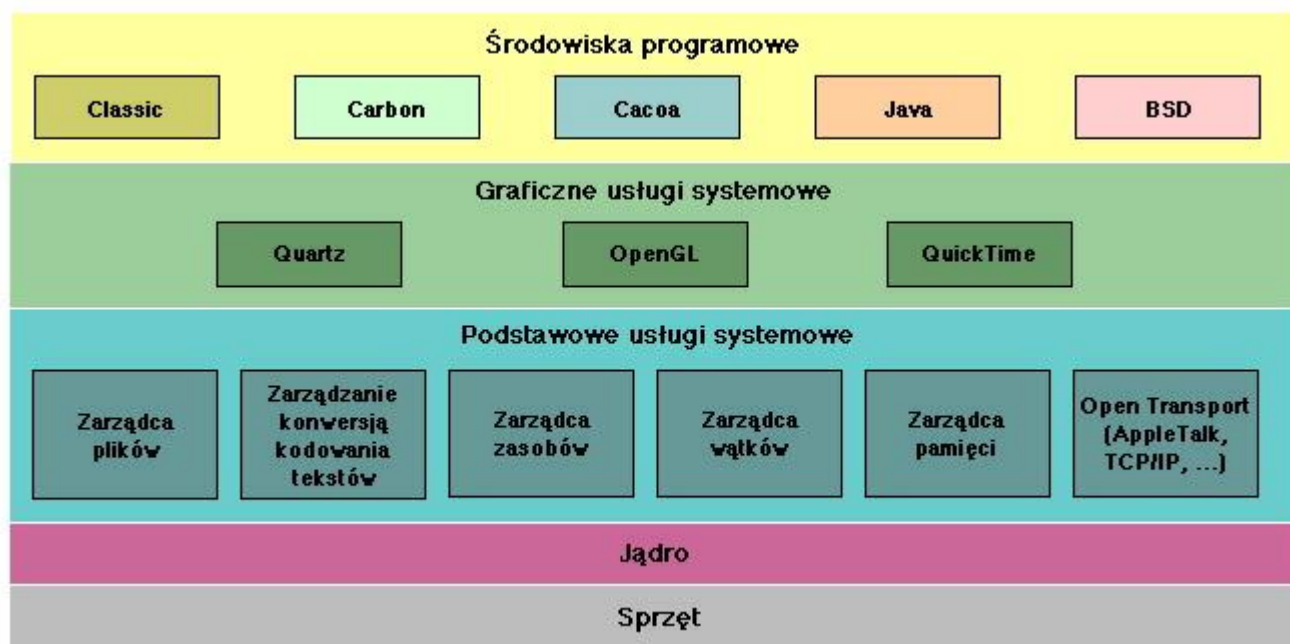
2.3. System Mac OS

W 1984 r. firma Apple przedstawiła pierwszą wersję systemu operacyjnego Mac OS, opracowanego specjalnie dla nowych komputerów osobistych Macintosh. Pod wieloma względami był to typowy system jednostanowiskowy, przeznaczony dla pojedynczego użytkownika. System nie oferował wielozadaniowości ani ochrony obszarów pamięci. Pod jednym względem był jednak wyjątkowo nowatorski. Był to pierwszy system operacyjny dla komputerów osobistych oferujący graficzny interfejs użytkownika, który przez długie lata wyznaczał w tym zakresie punkt odniesienia dla innych systemów operacyjnych. W kolejnych wersjach systemu dokonywano wielu ulepszeń, wprowadzając m.in. możliwość uruchamiania wielu programów, pamięć wirtualną, obsługę różnych protokołów sieciowych.

Najnowsza wersja systemu **Mac OS X** w pełni zasługuje na miano nowoczesnego systemu operacyjnego. Większość systemu została stworzona od nowa przy zachowaniu zalet wygodnego interfejsu graficznego oraz możliwości wykonywania starszych programów. Mac OS X zapewnia obecnie:

- wielozadaniowość,
- pamięć wirtualną,
- pełną ochronę obszarów pamięci,
- wieloprzetwarzanie symetryczne z wykorzystaniem wątków,
- przetwarzanie w łagodnym czasie rzeczywistym,
- wielodostęp.

Na Rys. 2.5 przedstawiono strukturę warstwową systemu.



Rys. 2.5 Struktura systemu Mac OS X

Jądro systemu wykorzystuje w dużym stopniu kod innych systemów operacyjnych, udostępniany swobodnie na zasadach otwartej licencji (Open Source), łącząc ze sobą następujące składowe:

- jądro systemu Mach 3.0,
- jądro systemu BSD 4.4,
- programy obsługi urządzeń i środowisko do ich tworzenia,

- obsługa różnych protokołów transmisji sieciowej,
- wirtualny system plików VFS obsługujący różne systemy plików.

Pełny kod jądra Mac OS X jest również oprogramowaniem swobodnie dostępnym pod nazwą [Darwin](#).

Dwie kolejne warstwy systemu zapewniają różnorodne usługi systemowe programom użytkowym. W wyższej warstwie zgrupowane są usługi środowiska graficznego systemu.

Mac OS X udostępnia kilka różnych środowisk programowych, czyli interfejsów programu użytkowego API:

- Classic - oryginalne środowisko starszej wersji systemu Mac OS 9,
- Carbon - adaptacja środowiska Mac OS 9 zoptymalizowana dla Mac OS X,
- Cocoa - nowe, zorientowane obiektowo środowisko systemu Mac OS X,
- Java - środowisko programów w języku Java,
- BSD - środowisko systemu BSD z tekstowym interpreterem poleceń,
- AppleScript - środowisko do tworzenia skryptów,
- WebObjects - środowisko do tworzenia aplikacji sieciowych.

W najnowszej wersji systemu środowiska Classic i Carbon nie są już wspierane.

2.4. System 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. System został jednak zaprojektowany zgodnie ze standardem POSIX 1003.1, który definiuje interfejsy programowania i narzędzi systemowych dla systemu UNIX. Dzięki temu zachowuje większość cech i możliwości systemu UNIX.

Wielu programistów z całego świata wciąż pracuje nad rozwojem jądra systemu oraz bogatego zestawu programów usługowych. Rozwój jądra jest koordynowany przez grupę programistów z Linusem Torvaldsem na czele. Przyjęto w tym celu następującą konwencję numeracji kolejnych wersji jądra:

x.y.z

gdzie:

x - numer wersji,

y - numer podwersji,

z - numer zbioru poprawek wprowadzonych do danej podwersji.

Pierwsza wersja systemu Linusa Torvaldsa nosiła oznaczenie 0.02. Wersja o numerze 1.0 pojawiła się w 1994 r. W momencie powstawania tego podręcznika najnowsze wersje jądra nosiły oznaczenia 4.3 (oraz 3.18.21 i 2.6.32.68 z poprzednich wersji, które wciąż stanowi podstawę wielu dystrybucji Linuksa).

Systemy Linux i UNIX mają podobną strukturę wewnętrzną (Rys. 2.1). Chociaż jądro Linuxa jest monolityczne, jego kod może być podzielony fizycznie na kilka części, łączonych dynamicznie w pamięci:

- jądro podstawowe, przechowywane w pamięci operacyjnej,
- moduły jądra, dołączane dynamicznie w czasie działania systemu.

Modularność jądra pozwala na elastyczne dostosowywanie jego funkcji do zmieniających się wymagań bez konieczności ponownej kompilacji całego kodu. Jądro podstawowe zajmuje poza tym znacznie mniej miejsca w pamięci operacyjnej, a moduły są ładowane do pamięci tylko wtedy, gdy są potrzebne. Moduły mogą zawierać programy obsługi nowych urządzeń podłączonych do systemu lub interfejsy programowe różnych typów systemów plików.

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.

System Linux jest rozpowszechniany w postaci pakietów oprogramowania, noszących nazwę dystrybucji, które zawierają:

- jądro systemu,
- zestaw bibliotek,
- zestaw programów systemowych,
- program instalacyjny,
- dokumentację w formie elektronicznej.

Dystrybucja łączy oprogramowanie różnych twórców. Wspólnym elementem wszystkich dystrybucji jest jądro systemu oraz pewien podstawowy zestaw programów systemowych i ich dokumentacji. Firmy przygotowujące dystrybucje dokonują wyboru dodatkowego oprogramowania i dokumentacji, opracowują własny program instalacyjny i ewentualnie dodatkową własną dokumentację. Do najpopularniejszych obecnie dystrybucji należą:

- Red Hat Enterprise,
- Fedora,
- CentOS,
- Scientific Linux,
- SuSE Linux Enterprise,
- openSuSE,
- Debian,
- Ubuntu,
- Slackware,
- Caldera,
- LinuxWare,
- Mandrake/Mandriva,
- Knoppix.

2.5. Sesja użytkownika w systemie Linux

Uruchamianie systemu

System Linux uruchamiany jest przez specjalny program ładujący o nazwie GRUB (ang. **GR**and **Un**ified **B**ootloader) w czasie startowania systemu komputerowego. Program ładuje do pamięci operacyjnej jądro Linuxa zapisane na dysku w pliku **vmlinuz** i przekazuje mu sterowanie systemem. Może również uruchamiać inne systemy operacyjne, zainstalowane równolegle na dysku. W takim przypadku daje użytkownikowi możliwość wyboru systemu operacyjnego, który chce uruchomić. Sposób działania programu ładującego określa plik konfiguracyjny **/boot/grub/grub.conf**. Można również używać starszego programu ładującego o nazwie LILO (ang. **L**inux **L**oader) z plikiem konfiguracyjnym **/etc/lilo.conf**.

Istnieje również możliwość uruchomienia systemu Linux przy pomocy dyskietki startowej, dysku CD/DVD lub urządzenia USB.

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 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** (lub **adduser**).

```
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]
```

Atrybuty użytkownika

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

Tablica 2.2 Atrybuty użytkownika

Atrybut	Znaczenie
nazwa	nazwa jednoznacznie identyfikująca konto użytkownika,
identyfikator użytkownika UID	numer jednoznacznie identyfikujący użytkownika w systemie,
identyfikator grupy GID	numer grupy, do której należy użytkownik,
katalog domowy	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**.

Terminale i konsole

System Linux stosuje podobny interfejs terminalu szeregowego, określany nazwą **tty**, do wszystkich urządzeń przeznaczonych do pracy interakcyjnej. Dotyczy to konsoli komputera, fizycznych terminali połączonych przez łącza szeregowo, X-terminali, czyli programów emulacji terminala w oknie systemu graficznego X Window (np. program **xterm**) oraz połączeń sieciowych. Terminale obsługujące okna systemu X Window i połączenia sieciowe nazywane są **pseudoterminalami**, w odróżnieniu od rzeczywistych terminali sprzętowych.

System może zwielokrotnić sesje terminala przy konsoli komputera, dla jednego ekranu i jednej klawiatury. Każda taka sesja, umożliwiająca rozpoczęcie pracy z systemem, nosi nazwę **konsoli wirtualnej** Linuxa. Konsola wirtualna ma własne ustawienia terminala i klawiatury. Przełączanie się pomiędzy konsolami umożliwiają specjalne sekwencje klawiszy. Na platformie sprzętowej PC stosuje się sekwencje **Ctrl+Alt+Fn**, gdzie **Fn** oznacza klawisze funkcyjne **F1**, **F2** ...

Interpreter poleceń

Podstawowym interpreterem poleceń w systemie Linux jest program o nazwie **bash**. Szczegółowy opis jego możliwości zawarto w wykładach 5 i 6.

Interpreter poleceń jest jednym z najważniejszych programów systemowych. Przyjmuje polecenia użytkownika wprowadzane w trybie tekstowym i wykonuje je bezpośrednio lub uruchamia w tym celu odpowiednie programy. Niektóre programy można uruchomić wyłącznie w taki sposób, zwłaszcza gdy wymagają podania opcji i argumentów wywołania. Nie da się tego zrobić z poziomu menu interfejsu graficznego.

Nowoczesne interpretery oferują użytkownikom bardzo bogate możliwości takie, jak:

- rozbudowana edycja poleceń,
- automatyczne uzupełnianie nazw programów i plików,
- rozwijanie złożonych nazw plików,
- zapamiętywanie i przywoływanie wydanych poleceń.

Środowisko graficzne X Window

Standardowym środowiskiem graficznym systemu UNIX jest system X Window. Jego aktualna wersja określana jest nazwą X11 Release 7. Linux dysponuje darmową wersją tego środowiska o nazwie XFree86, która zachowuje pełną zgodność z wersją X11.

System X Window tworzy sieciowe środowisko graficzne oparte na modelu klient-serwer. Aplikacje serwera i klientów mogą działać na różnych komputerach. Umożliwia to uruchomienie programu na jednym komputerze i wyświetlanie jego wyników w oknie graficznym na drugim komputerze.

Aplikacje klientów, działające lokalnie lub na zdalnych komputerach, korzystają z możliwości graficznych serwera do wyświetlania informacji na ekranie monitora. Serwer systemu, określany jako X-serwer, działa na lokalnym komputerze i spełnia następujące zadania:

- na żądanie programów klientów wykonuje operacje graficzne na konsoli komputera: wyświetla okna, ikony, kształty, tekst,
- przekazuje dane wejściowe z klawiatury i myszki lokalnej konsoli do właściwego programu klienta.

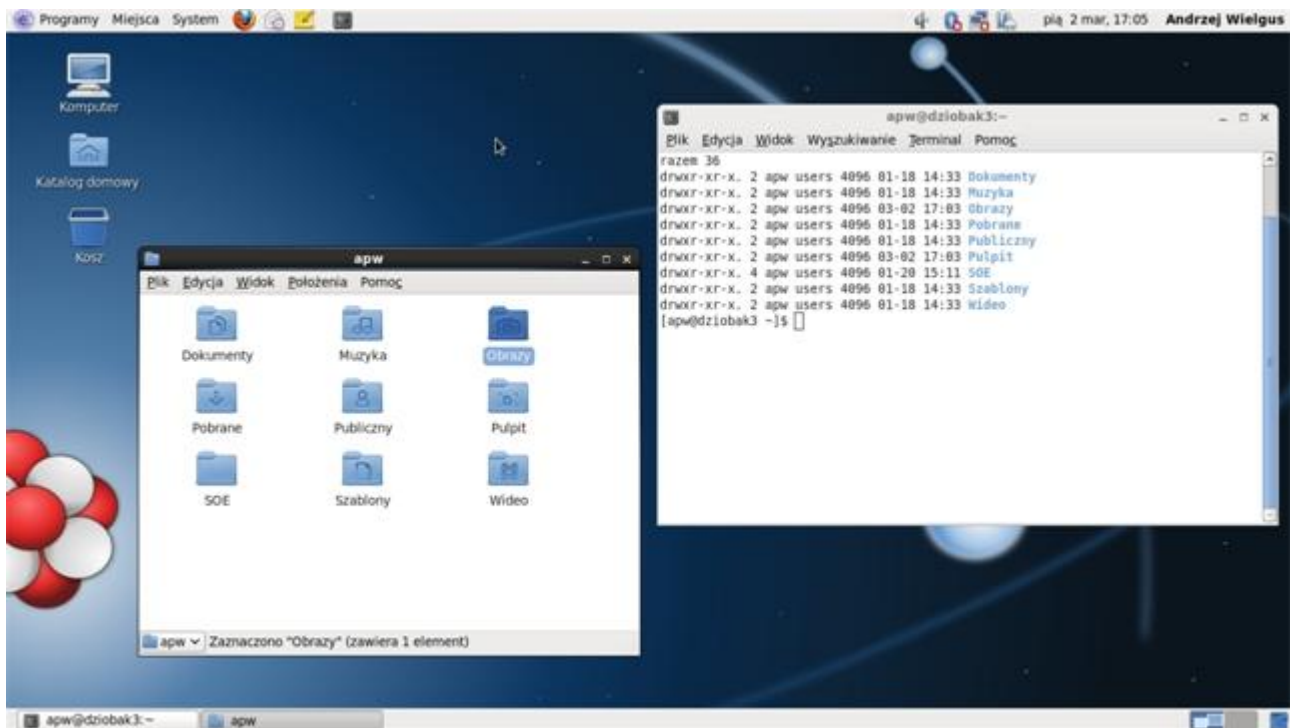
Po zalogowaniu do systemu użytkownik pracujący przy konsoli komputera może uruchomić środowisko graficzne poleceniem **startx**. System operacyjny uruchamia wtedy serwer systemu X Window o nazwie **X**, dostosowany do parametrów karty graficznej komputera.

Uruchamianiem i zarządzaniem oknami w środowisku graficznym zajmuje się program menedżera okien (ang. *window manager*). Decyduje on o wyglądzie wyświetlanych okien: kolorze, czcionkach, kształcie przycisków, wyglądzie menu. Interpretuje również ruchy myszką i naciśnięcia przycisków. Do najpopularniejszych menedżerów należą: **twm**, **fvwm**, **fvwm2**, **mwm**, **kwm**, **sawfish**.

Obecnie coraz większą popularność zdobywają systemy aktywnego pulpitu (ang. *active desktop environment*) takie, jak KDE czy GNOME. Oprócz menedżera okien w ich skład wchodzi wiele innych programów. Dzięki temu oferują wygodne środowisko graficzne użytkownika, na które składa się zwykle:

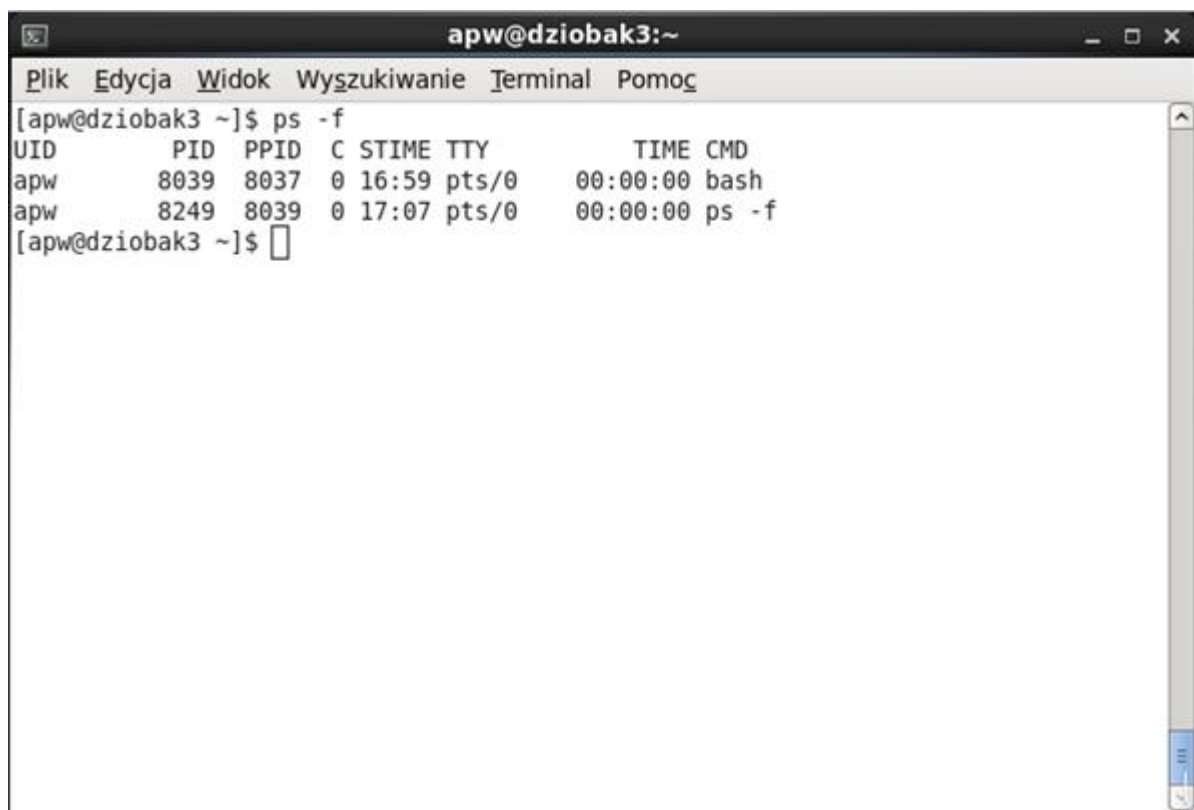
- pulpit przeznaczony do umieszczania okien działających programów, a także ikon (skrótów) ułatwiających ich uruchamianie,
- panel narzędziowy służący do uruchamiania programów (poprzez wybór z rozwijanego menu lub wybór symbolu za pomocą myszki) oraz do wyświetlania statusu,
- zestaw standardowych programów pulpitowych, jak emulator terminala, menedżer plików, przeglądarka dokumentacji systemu, konfigurator wyglądu okien i pulpitu, zegar, kalkulator itp.

Systemy aktywnego pulpitu zorientowane są na pracę z plikami umożliwiając ich łatwe przenoszenie między programami i pulpitem. Rys. 2.6 prezentuje przykładowy wygląd pulpitu GNOME.



Rys. 2.6 System aktywnego pulpitu GNOME

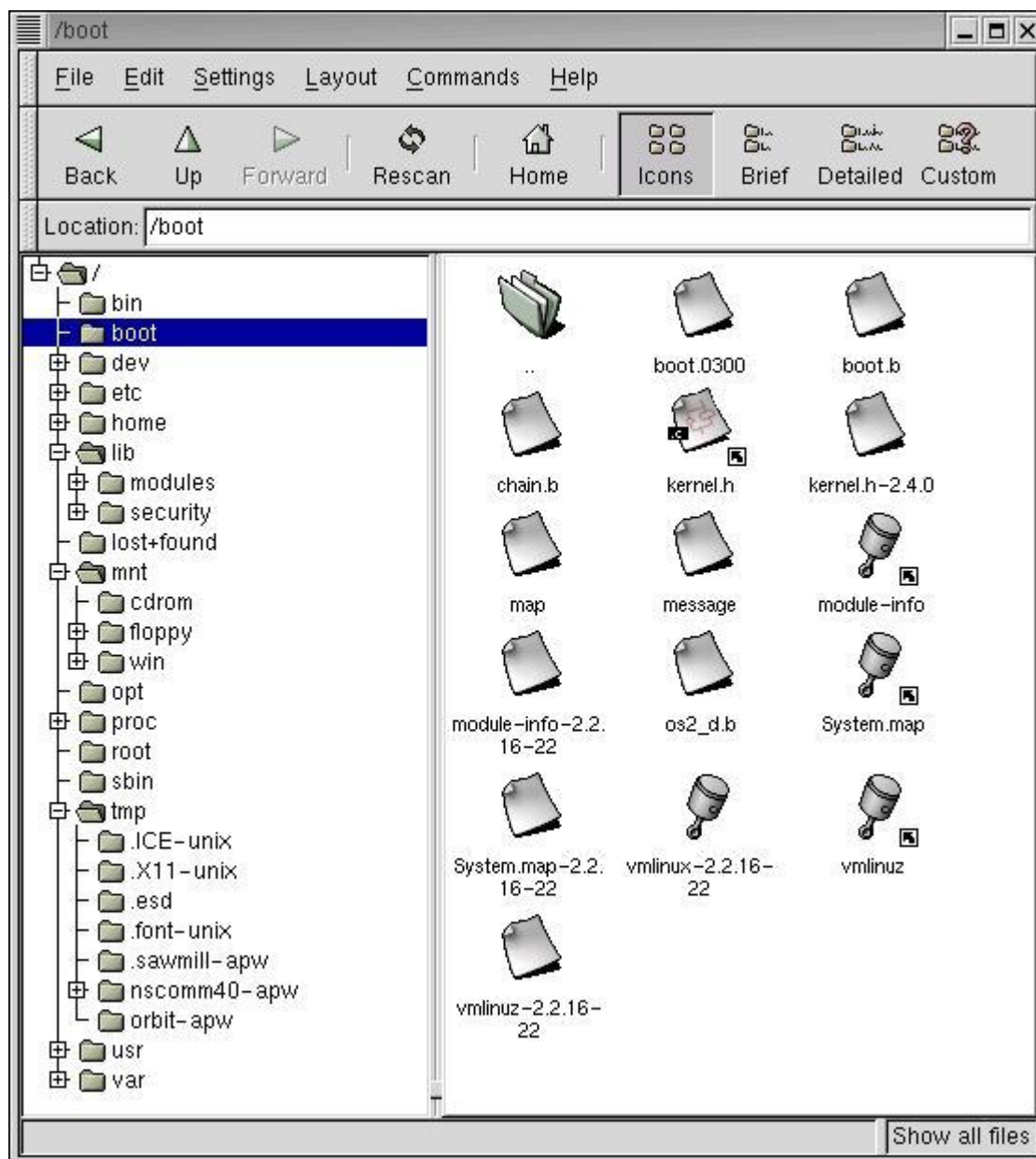
Jednym z najważniejszych programów jest program emulacji terminala, który pozwala otworzyć okno z uruchomionym interpreterem poleceń. Użytkownik uzyskuje w ten sposób możliwość uruchamiania programów, które nie korzystają wprost z okienkowego interfejsu graficznego. Najpopularniejszym emulatorem terminala jest obecnie program **xterm**, który stał się praktycznie standardem we wszystkich wersjach systemu UNIX i wszystkich dystrybucjach Linuksa.



Rys. 2.7 Okno emulatora terminala

Systemy oferują na ogół również inne emulatory terminala 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**. Na Rys. 2.7 prezentujemy przykładowe okno tego programu z uruchomionym interpreterem **bash**.

Innym przydatnym programem jest graficzny menedżer plików, który umożliwia oglądanie zawartości katalogów, manipulację plikami, przeglądanie i edycję zawartości plików. Takimi programami są między innymi GNU Midnight Commander oraz Nautilus, pokazany na Rys. 2.8.



Rys. 2.8 Program menedżera plików

Kończenie sesji i zamykanie systemu

Zakończenie sesji użytkownika, czyli popularnie **wylogowanie**, następuje w wyniku wydania polecenia **logout** lub **exit** w powłoce logowania. Należy jednak pamiętać, że nie oznacza to bezwzględnego zakończenia wszystkich procesów użytkownika. Niektóre procesy mogą pozostać w systemie i kontynuować swoje działanie niezależnie od zakończenia sesji.

Przed wyłączeniem komputera należy zamknąć system Linux w bezpieczny sposób. Wymaga to użycia jednego z poleceń: **shutdown** lub **halt**. Polecenie **reboot** umożliwia natomiast bezpieczne zamknięcie systemu i jego ponowne uruchomienie, czyli **restart** systemu. Powyższe polecenia może wykonać wyłącznie użytkownik uprzywilejowany **root**.

```
/sbin/shutdown [-t sek] [-opcje] czas [wiadomość]
```

gdzie:

- t sek** - opóźnienie w sekundach pomiędzy wysłaniem ostrzeżenia do wszystkich procesów a zamknięciem systemu,
- czas** - czas zamknięcia systemu,
- wiadomość** - ostrzeżenie o zamknięciu systemu wysyłane do wszystkich zalogowanych użytkowników.

Opcje pozwalają określić szczegółowo, jakie działanie ma zostać wykonane.

Przykład

- shutdown -h now** - powoduje natychmiastowe zamknięcie systemu,
- shutdown -r now** - powoduje natychmiastowy restart systemu.

Na platformie sprzętowej PC Linux wykonuje również bezpieczny restart komputera obsługując typową sekwencję klawiszy **Ctrl+Alt+Del**.

W przypadku korzystania ze środowiska graficznego powłoka logowania nie jest bezpośrednio dostępna. Możliwe są wtedy dwie drogi postępowania:

- zakończenie sesji graficznej i wydanie odpowiedniego polecenia w powłoce logowania,
- wykorzystanie poleceń wybieranych z odpowiedniego menu - zazwyczaj są to pozycje: **Logout**, **Reboot** i **Halt** lub **Shutdown**.

Dokumentacja w formie elektronicznej

Dokumentacja do systemu Linux występuje w większości w postaci elektronicznej. Największy zbiór dokumentów stanowi projekt dokumentacji Linuksa [LDP](#) (ang. Linux Documentation Project), na który składają się następujące elementy:

- pliki pomocy w formacie **man** (ang. *manual pages*, *man pages*),
- dokumenty w formacie **Info**,
- dokumenty HOWTO (w języku angielskim),
- książki LDP.

Standardową dokumentację programów oraz funkcji systemowych i bibliotecznych w systemach UNIX i Linux stanowią pliki pomocy o dostępie bezpośrednim. Korzystanie z nich umożliwia program **man**.

```
man [opcje] [sekcja] nazwa
```

gdzie:

nazwa - nazwa programu lub funkcji,

sekcja - numer sekcji.

Pliki pomocy podzielone są na sekcje w zależności od tematyki, jakiej dotyczą. Jeżeli podany jest numer sekcji, to program **man** przeszukuje tylko tą sekcję dokumentacji. W przeciwnym przypadku przeszukiwane są kolejno wszystkie sekcje. Podanie numeru sekcji może okazać się konieczne, ponieważ w kilku sekcjach mogą znajdować się dokumenty o takiej samej nazwie, ale opisujące inne tematy.

Pomocne okazać mogą się również programy **apropos** i **whatis**, które wyszukują słowa kluczowe w bazie danych zawierających krótkie opisy plików pomocy.

```
apropos słowo_kluczowe ...
```

```
whatis słowo_kluczowe ...
```

Dokumenty w formacie **Info** można przeglądać przy pomocy programu **info**, działającego w trybie interakcyjnym.

```
info [opcje]
```

Dokumenty [HOWTO](#) (w języku angielskim) dostępne są w postaci plików tekstowych. Zgodnie z ich nazwą zawierają opisy rozwiązywania konkretnych problemów występujących w systemie Linux.

Podręczniki LDP dostępne są w formacie PDF i HTML. Mają postać obszernych opracowań dotyczących administrowania, programowania i użytkowania systemu Linux oraz opisu jego budowy wewnętrznej.

Bibliografia

1. Silberschatz A., Galvin P.B.: Podstawy systemów operacyjnych, WNT 2000 (rozdział 22)
2. Glass G., Ables K.: Linux dla programistów i użytkowników, Wydawnictwo Helion 2007 (rozdziały: 1, 2)