

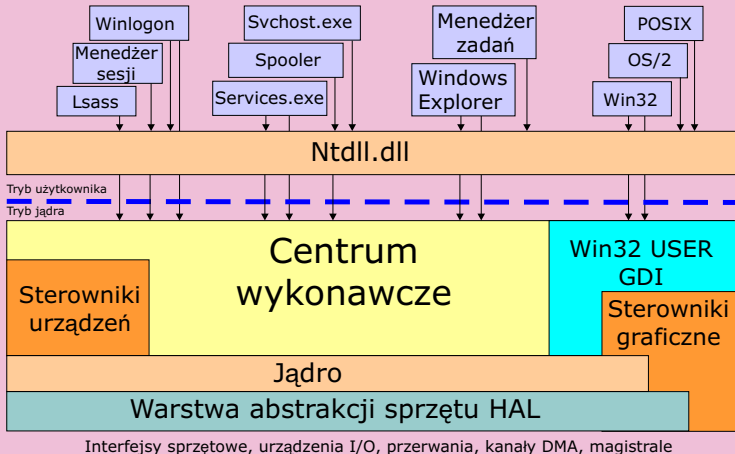
Przegląd popularnych systemów operacyjnych

dr hab. inż. Krzysztof Patan, prof. PWSZ

Instytut Politechniczny
Państwowa Wyższa Szkoła Zawodowa w Głogowie
k.patan@issi.uz.zgora.pl

System Windows 2000

Struktura systemu



Tryb jądra i tryb użytkownika

- **Tryb jądra** – uprzywilejowany tryb wykonywania kodu, w którym procesor zezwala na dostęp do całej pamięci systemu oraz do wszystkich instrukcji procesora
- W trybie jądra działa kod systemu operacyjnego
- **Tryb użytkownika** – nieuprzywilejowany tryb procesora, w którym działają aplikacje użytkownika
- W trybie użytkownika dostępna jest ograniczona liczba interfejsów oraz ograniczony dostęp do zasobów systemowych
- Dwa tryby pracy procesora wprowadzono w celu ochrony krytycznych dla systemu operacyjnego danych
- Mechanizm uniemożliwiający błędnej aplikacji użytkownika naruszenie stabilności całego systemu

Komponenty systemu Windows 2000

☛ Warstwa uniezależnienia od sprzętu

- warstwa uniezależnienia od sprzętu HAL (ang. *Hardware Abstraction Layer*)
- HAL odgrywa zasadniczą rolę w zapewnieniu przenaszalności systemu operacyjnego
- jest zewnętrznym pracującym w trybie jądra modułem (`hal.dll`)
- zapewnia niskopoziomowy interfejs do platformy sprzętowej
- wewnętrzne komponenty systemu oraz sterowniki urządzeń chcąc dostać się do sprzętu wywołują procedury warstwy HAL
- podczas instalacji systemu wybierana jest odpowiednia dla danej platformy sprzętowej wersja warstwy HAL

☛ Jądro

- zawiera zestaw funkcji zapewniających podstawowe mechanizmy (szeregowanie wątków, synchronizacja) wykorzystywane przez centrum wykonawcze oraz niskopoziomowe funkcje zależne od platformy sprzętowej (obsługa przerw i wyjątków)
- kod jądra napisano w C/C++ z wstawkami asemblerowymi (specjalizowane instrukcje procesora oraz rejestrów)
- dolna warstwa pliku `ntoskrnl.exe` lub `ntkrnlpa.exe` (dla obsługi pamięci większej niż 4 GB) (folder `System32`)
- Windows 2000 posiada jądro monolityczne – większość kodu systemu operacyjnego współdzieli tę samą, chronioną przestrzeń adresową trybu jądra
- istnieje możliwość zakłócenia integralności danych jednego komponentu systemu przez inny

☛ Sterowniki urządzeń

- są zewnętrznymi modułami jądra (pliki z rozszerzeniem `.sys`), które służą jako interfejs pomiędzy menedżerem I/O, a odpowiednimi urządzeniami
- działają w trybie jądra w jednym z trzech kontekstów:
 - 1 w kontekście wątku użytkownika, który zainicjował funkcję I/O
 - 2 w kontekście systemowego wątku trybu jądra
 - 3 jako rezultat przerwania
- sterowniki nie wykonują operacji bezpośrednio na sprzęcie, lecz za pośrednictwem warstwy HAL
- zainstalowanie sterownika jest jedynym sposobem dodania do systemu (wykonywanego w trybie jądra) kodu stworzonego przez użytkownika – możliwość dostępu do wewnętrznych funkcji lub struktur danych systemu operacyjnego

☛ Centrum wykonawcze

- górna warstwa pliku `ntoskrnl.exe` (`ntkrnl0pa.exe`)
- na centrum wykonawcze składają się:
 - ◇ Menedżer konfiguracji – implementacja i zarządzanie rejestrem systemowym
 - ◇ Menedżer procesów i wątków – funkcje systemowe do obsługi procesów i wątków zaimplementowano w jądrze, natomiast centrum nadaje im funkcjonalność
 - ◇ Monitor bezpieczeństwa odwołań – ochrona zasobów systemu operacyjnego
 - ◇ Menedżer pamięci podręcznej – zwiększa wydajność plikowych operacji I/O poprzez składowanie w pamięci danych dyskowych do których dostęp odbywał się niedawno; opóźnianie operacji zapisu, poprzez składowanie uaktualnień w pamięci przed wysłaniem ich na dysk
 - ◇ Menedżer energii – koordynuje zdarzenia związane z poborem mocy przez system; generuje komunikaty dotyczące zarządzania energią; gdy system nie wykonuje obliczeń może przewidywać obniżenie poboru mocy poprzez uśpienie procesora

- ◇ Menedżer PnP – określa i ładuje sterowniki potrzebne do obsługi konkretnych urządzeń; pobiera wymagania zasobów sprzętowych każdego urządzenia; na podstawie wymagań sprzętowych urządzeń przyznaje odpowiednie zasoby (porty I/O, przerwania IRQ, kanały DMA, adresy pamięci); wysyła komunikaty w przypadku zmiany stanu urządzeń
- ◇ Menedżer I/O – implementacja niezależnych od sprzętu operacji I/O i przekierowanie żądań do odpowiednich sterowników urządzeń
- ◇ Menedżer pamięci wirtualnej – implementuje pamięć wirtualną zapewniającą każdemu procesowi dużą, prywatną przestrzeń adresową, która może przekraczać dostępną przestrzeń fizyczną
- ◇ Menedżer obiektów – tworzy, zarządza i usuwa obiekty Windows 2000 oraz abstrakcyjne typy danych używane do reprezentacji zasobów systemu operacyjnego (procesy, wątki obiekty synchronizacyjne)
- ◇ Mechanizm LPC (ang. *Local Procedure Call*) – przekazuje komunikaty między procesem klienta, a procesem serwera na tym samym komputerze; jest to wersja zdalnych wywołań procedur (ang. *Remote Procedure Call* – RPC)

👉 Ntdll.dll

- specjalna biblioteka wspierająca system; wspomaga korzystanie z bibliotek podsystemowych
 - ◇ procedury przekazujące wywołania usług systemowych do centrum wykonawczego
interfejs do wewnętrznych usług Windows 2000 (ponad 200)
 - ◇ wewnętrzne funkcje pomocnicze, wykorzystywane przez podsystemy, biblioteki podsystemów i inne komponenty

Procesy systemowe

Proces jąłowy

- proces nie wykonuje żadnego programu trybu użytkownika
- zawiera po jednym wątku dla każdego procesora; wątek ten jest wykonywany w jąłowym czasie procesora
- proces posiada identyfikator 0

Proces systemowy

- identyfikator procesu – zawsze 8
- zawiera wątki specjalnego rodzaju wykonujące się jedynie w trybie jądra
- wątki systemowe są tworzone przez system oraz różne sterowniki urządzeń podczas inicjacji systemu

Menedżer sesji `smss`

- lokalizacja: `\winnt\system32\smss.exe`
- pierwszy proces trybu użytkownika w systemie
- jest odpowiedzialny za czynności podczas uruchamiania systemu (otwarcie plików stronicowania, operacje zmiany nazwy i usunięcia plików, inicjacja systemowych zmiennych środowiskowych)
- uruchamia procesy podsystemów (`csrss.exe`) oraz proces `winlogon`, który tworzy pozostałe procesy w systemie
- po inicjacji główny wątek w `smss` oczekuje na uchwytach do procesów `csrss` i `winlogon`; jeśli którykolwiek z tych procesów nieoczekiwanie zakończy się, `smss` powoduje załamanie systemu
- oczekuje także na żądania załadowania podsystemów oraz zdarzenia związane z wykrywaniem błędów

Proces logowania winlogon

- lokalizacja `\winnt\system32\winlogon.exe`
- obsługuje logowanie i wylogowanie się użytkowników
- proces dokonuje częściowej inicjacji środowiska użytkownika (ustanowienie liter napędów, polityka bezpieczeństwa)
- tworzy proces, w którym uruchamiana jest powłoka systemu `explorer.exe`
- proces przechwytuje sekwencję SAS (ang. *Secure Attention Sequence*) z klawiatury (Ctrl+Alt+Delete); może w ten sposób uruchomić Menedżera zadań, zablokować komputer lub zamknąć system

Lokalny serwer bezpieczeństwa LSASS

- lokalizacja: `\winnt\system32\lsass.exe`
- otrzymuje żądania uwierzytelniania od procesu winlogon i wywołuje odpowiedni pakiet uwierzytelniający celem dokonania weryfikacji
- po uwierzytelnieniu, proces generuje tzw. *znacznik dostępu* zawierający profil bezpieczeństwa użytkownika

Menedżer kontroli usług SCM

- lokalizacja: `\winnt\system32\services.exe`
- specjalny proces systemowy odpowiadający za uruchamianie, zatrzymywanie i interakcje z procesami usługowymi
- usługi mogą odnosić się do procesów serwerowych lub do sterowników
- usługi mogą być uruchamiane automatycznie lub ręcznie
- zainstalowane usługi można wyświetlić za pomocą narzędzia Usługi znajdującego się w Narzędziach administracyjnych

Rejestr systemowy

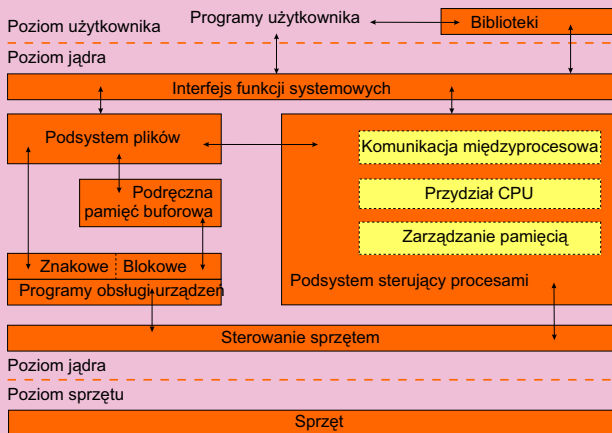
- kluczowy komponent w konfiguracji i zarządzaniu systemem
- rejestr jest bazą danych składającą się z kluczy oraz wartości
- rejestr ma strukturę drzewa z sześcioma kluczami głównymi:
 - 1 HKEY_CURRENT_USER – dane dotyczące aktualnie zalogowanego użytkownika
 - 2 HKEY_USERS – przechowuje informacje o wszystkich kontach dostępnych na danym komputerze
 - 3 HKEY_CLASSES_ROOT – dane o skojarzeniu plików z aplikacjami
 - 4 HKEY_LOCAL_MACHINE – dane dotyczące systemu
 - 5 HKEY_PERFORMANCE_DATA – dane dotyczące wydajności
 - 6 HKEY_CURRENT_CONFIG – informacje o aktualnym profilu sprzętowym
- aplikacje do pracy z rejestrem regedit i regedt32

Pliki zawierające części rejestru

Ścieżka w rejestrze	Ścieżka w systemie plików
HKEY_LOCAL_MACHINE\SYSTEM	\winnt\system32\config\system
HKEY_LOCAL_MACHINE\SAM	\winnt\system32\config\sam
HKEY_LOCAL_MACHINE\SECURITY	\winnt\system32\config\security
HKEY_LOCAL_MACHINE\SOFTWARE	\winnt\system32\config\software
HKEY_LOCAL_MACHINE\HARDWARE	ulotny
HKEY_LOCAL_MACHINE\SYSTEM\Clone	ulotny
HKEY_USERS\ <security id="" of="" td="" username><=""><td>\documents and settings\<username>\ntuser.dat</username></td></security>	\documents and settings\ <username>\ntuser.dat</username>
HKEY_USERS\ <security id="" of="" td="" username>_classes<=""><td>\documents and settings\<username>\local settings\application data\microsoft\windows\usrclass.dat</username></td></security>	\documents and settings\ <username>\local settings\application data\microsoft\windows\usrclass.dat</username>
HKEY_USERS\.DEFAULT	\winnt\system32\config\default

System UNIX

Struktura jądra



Plik i proces – podstawowe pojęcia w modelu systemu

Jądro (ang. *kernel*)

Program posiadający podczas działania systemu operacyjnego szereg przywilejów dostępu do zasobów fizycznych komputera

Usługi jądra

- 1 zarządzanie jednostkami pamięci
- 2 określanie zawartości rejestrów procesora
- 3 pozwala innym procesom, serwerom na kontakt z zasobami fizycznymi komputera
- 4 określa przestrzenie adresowe do wzajemnej ochrony procesów przed sobą
- 5 nadzorowanie mechanizmem przywoływania zasobów zarządzanych przez jądro - pułapki systemowe (ang. *system call trap*)

Jądra monolityczne i mikrojądra




1 Jądro monolityczne

- jądro monolityczne wykonuje wszystkie podstawowe funkcje systemu operacyjnego
- rozmiary jądra sięgają megabajta kodu i danych
- jądro umieszczone jest zazwyczaj w pliku `/unix` (`/vmunix` `/vmlinuz` `/genunix`)
- jądro jest nierozbieralne - budowa niemodularna
- jądro nie jest podatne na modyfikacje
- dostosowanie jakiegokolwiek elementu jego oprogramowania do zmieniających się wymagań jest trudne

- jądro monolityczne może zawierać pewną liczbę procesów usługowych, które działają w jego ramach (serwery plików, procesy sieciowe)
- wiele usług świadczonych przez jądro może być dostarczanych przez usługi otwarte

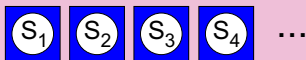


jądro monolityczne

-  program ładowany dynamicznie
-  kod i dane jądra
-  serwer, program usługowy

2 Mikrojądro (ang. *microkernel*)

- mikrojądro to jądro realizujące najmniejszy z możliwych zbiorów usług i zasobów
- wszystkie inne usługi systemowe są dostarczane przez serwery umieszczane dynamicznie tylko w tym komputerach systemu rozproszonego, które mają dostarczać danych usług



mikrojądro

- rozmiary mikrojądra wahają się od 10 KB do kilkuset KB
- mikrojądro jest tak zaprojektowane aby można je było przenosić między różnymi architekturami komputerów

- większa część kodu zaprojektowana jest w języku wysokiego poziomu np. C lub C++
- właściwości jądra ułożone są warstwowo, tak aby składniki zależne od maszyny redukowały się do minimalnej dolnej warstwy
- budowa typowego mikrojądra



Zarządca procesów – zajmuje się tworzeniem procesów i wykonywaniem na nich niskopoziomowych operacji

Zarządca wątków – odpowiada za tworzenie wątków, ich synchronizację oraz planowanie. Wątki to działania skojarzone z procesem

Zarządca komunikacji – zapewnia łączność między wątkami należącymi do różnych procesów lokalnych. Można tutaj umieścić poziom komunikacji pomiędzy wątkami procesów zdalnych

Zarządca pamięci – administruje zasobami pamięci fizycznej, jednostką zarządzania pamięcią oraz sprzętowymi pamięciami podręcznymi

Nadzorca – koordynuje obsługę przerwań, wywołań pułapek systemowych i innych wyjątków

Porównanie

- Mikrojądro
 - ✧ otwartość
 - ✧ małe rozmiary
 - ✧ mniejsza podatność na błędy
- Jądro monolityczne
 - ✧ duże rozmiary
 - ✧ trudne w pielęgnacji
 - ✧ moduły jądra wykonywane są w tej samej przestrzeni adresowej
 - ✧ istnieje możliwość naruszania ścisłych reguł modularności
 - ✧ błąd w jednym module może źle wpływać na działanie innego modułu
 - ✧ ponowna implementacja modułu pociąga za sobą przebudowę jądra
 - ✧ dobra wydajność wywoływania operacji

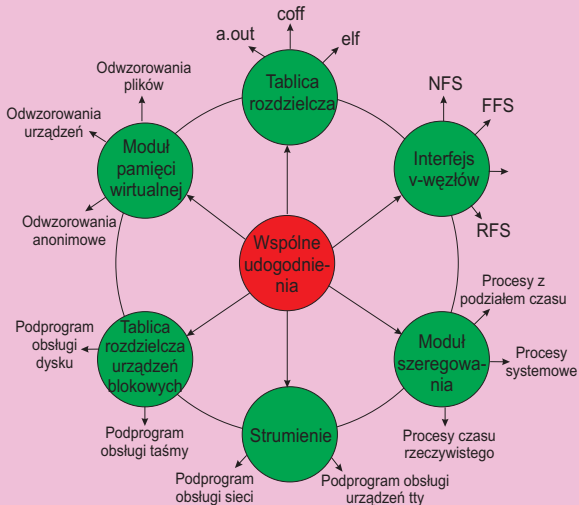
- 1 **Funkcje systemowe** – zbiór funkcji systemowych podzielono na dwie grupy:
 - funkcje systemowe do komunikacji procesów z podsystemem plików: `open` , `close` , `read` , `write` , `stat` , `chown` , `chmod`
 - funkcje systemowe do sterowania procesami: `fork` , `exec` , `exit` , `wait` , `brk` , `signal`
- 2 **Podsystem plików** – zarządza plikami, przydziela na nie pamięć, administruje pamięcią wolną, steruje dostępem do plików, udostępnia użytkownikom dane
- 3 **Podręczna pamięć buforowa** – podsystem plików udostępnia dane z plików za pomocą mechanizmu buforowania, regulacja przepływu danych między jądrem i urządzeniami pamięci pomocniczej
- 4 **Programy obsługi urządzeń** – są to moduły jądra sterujące pracą urządzeń peryferyjnych
 - blokowe – urządzenia o dostępie bezpośrednim lub urządzenia dla których ich podprogramy tworzą taką iluzję
 - znakowe (surowe) – reszta urządzeń, które nie są blokowe

- 6 **Podsystem sterowania procesami** – odpowiada za synchronizację procesów, komunikację między nimi, współpracuje z podsystemem plików podczas ładowania pliku do pamięci w celu wykonania
- **zarządzanie pamięcią** – steruje przydziałem pamięci (wymiana i stronicowanie na żądanie)
 - **przydział CPU** – przydzielanie czasu procesora procesom gotowym do wykonania, ustalanie kolejności wykonywania procesów
 - **komunikacja międzyprocesowa** – realizowane są różne sposoby komunikacji międzyprocesowej, asynchroniczne sygnalizowanie zdarzeń i synchroniczna transmisja komunikatów między procesami
- 7 **Sterowanie sprzętem** – odpowiada za obsługę przerw i za komunikowanie się z komputerem. Dyski czy terminale mogą przerywać pracę jednostki centralnej podczas wykonywania procesu. Jądro może wznowić wykonywanie przerwanej procesu po zakończeniu obsługi przerwania

Podsumowanie

- klasyczne jądro uniksowe było monolityczne i nie dawało się łatwo rozszerzać
- w miarę wzbogacania funkcjonalności systemu stawało się coraz większe i bardziej złożone
- jednym z rozwiązań była architektura mikrojądra, np. system Mach
- wydajność mikrojądra okazała się nieporównywalna z wydajnością tradycyjnego jądra
- więcej korzyści przyniosły próby modularyzacji i ładowania dynamicznego

Budowa nowoczesnego jądra



System Linux

- System Linux sprawia wrażenie systemu podobnego od innych systemów uniksowych
- Zaprojektowano go tak, aby był podobny do Uniksa i w pełni zgodny z Uniksem

Definicja systemu

Linux jest swobodnie dostępnym jądrem systemu operacyjnego opartego na systemie UNIX

Spojrzenie całościowe

Linux jest swobodnie dostępnym jądrem systemu operacyjnego opartego na Uniksie, który zawiera jądro, narzędzia systemowe, programy użytkowe i pełne środowisko do pisania i modyfikowania programów

Powstanie i rozwój systemu Linux

● 1991

- ◇ Linus Torvalds, student Uniwersytetu w Helsinkach rozpoczyna projekt dotyczący programowania niskopoziomowego procesora Intel 80386
- ◇ opiera się na systemie Minix (Andrew Tanenbaum)
- ◇ powstaje jądro systemu Linux 0.01
- ◇ autor początkowo obłożył system restrykcyjną licencją, ale później zdecydował się na licencję GPL (ang. *General Public License*)
 - Linux nie jest oprogramowaniem *public domain*
 - Linux – oprogramowanie w wolnym obiegu
 - zakaz prywatyzacji pochodnych produktów Linuksa
 - oprogramowania na zasadach licencji GPL nie wolno rozsyłać w formie czysto binarnej
- ◇ jądro zostało udostępnione w sieci Internet – burzliwy rozwój systemu

Projekt Linux

- system operacyjny Linux składa się z jądra systemu i innych składowych
- jądro systemu jest zestawione w całości przez kod napisany od początku specjalnie dla projektu Linux
- znaczna część pomocniczego oprogramowania nie należy wyłącznie do systemu Linux, lecz jest wspólna dla wielu systemów uniksowych – wiele narzędzi systemu BSD z Berkeley, system X Window z MIT, oprogramowanie GNU
- jako całość Linux jest utrzymywany przez luźną sieć twórców współpracujących za pomocą Internetu
- społeczność linuksowa utrzymuje dokument pt. Standard Hierarchii Systemu Plików w celu zachowania zgodności między różnymi składowymi systemu
- ważne cele projektowe – szybkość i wydajność
- standaryzacja – Linux pozostaje w zgodzie z istotnymi opisami standardu POSIX

Upowszechnianie systemu

- system Linux może zainstalować każdy, sprowadzając najnowsze wersje niezbędnych części systemu ze stanowisk ftp i kompilując je
- z biegiem czasu osoby i grupy osób dołożyły starań, by instalacja była jak najmniej skomplikowana
- dostarczano zawnazu skompilowane zbiory pakietów do łatwego instalowania nazywane dystrybucjami
- pierwszą uznaną dystrybucją był zbiór pakietów nazwany SLS – brak narzędzi zarządzania pakietami
- dystrybucja Slackware – istotne ulepszenie
- popularna dystrybucja Red Hat – upowszechniana komercyjnie
- Debian – dystrybucja rozpowszechniana bezpłatnie przez społeczność linuksową
- inne znane dystrybucje – Caldera, CraftWorks, Mandrake
- różnorodność różnych dystrybucji nie przeszkadza w ich wzajemnej zgodności

Zasoby internetowe

❑ serwisy WWW na świecie

- slashdog.org
- linux.org
- portalux.com
- www.linuxdoc.org
- linuxjurnal.com
- linuxword.com
- linuxgazette.com
- linuxtoday.com
- lwn.net

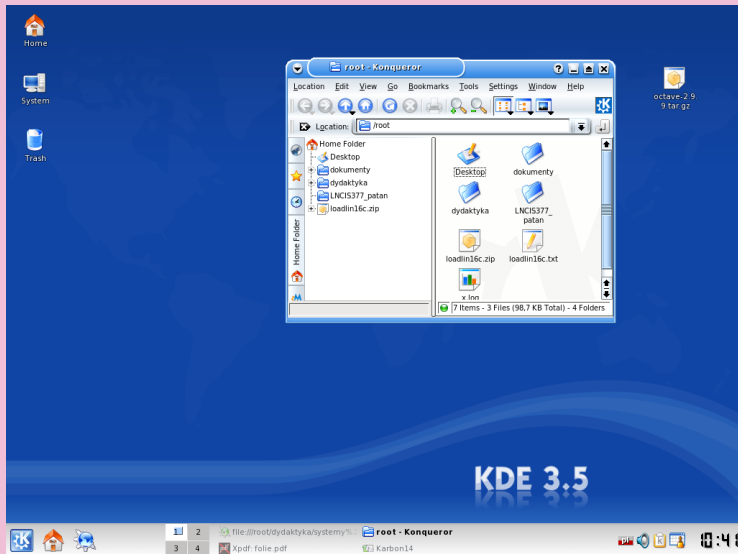
❑ polskie strony WWW

- linux.org.pl
- jzt.org.pl
- ultra.wsp.krakow/linuxpl
- linux.com.pl
- www.pld.org.pl

❑ listy adresowe

❑ grupy dyskusyjne

Linux - pulpit

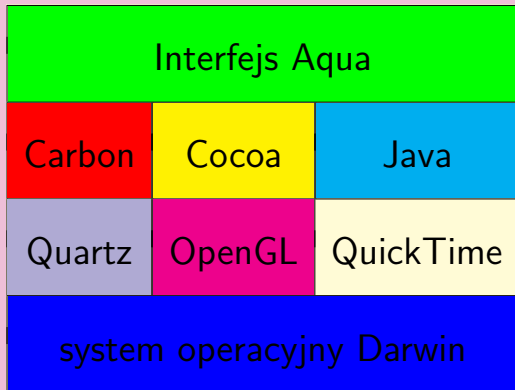


Mac OS X

Złożony system operacyjny dla komputerów Macintosh opracowany w roku 2000 w amerykańskiej firmie Apple Computer. X w nazwie oznacza 10 wersję systemu Mac OS, jednak Mac OS X jest systemem o całkowicie odmiennej budowie niż poprzednie wersje Mac OS.

- Mac OS X powstał w oparciu o mikrojądro Mach oraz usługi i narzędzia zaczerpnięte z projektów NetBSD oraz FreeBSD
- Podstawą systemu jest opracowany w Apple Computer system operacyjny Darwin
- Graficzny interfejs użytkownika nosi nazwę Aqua i jest własnym rozwiązaniem opracowanym przez Apple z wykorzystaniem doświadczeń i wzorów z poprzednich systemów tej firmy
- Apple udostępnia także własną wersję X Window System

Struktura systemu



Właściwości

- możliwość uruchamiania programów napisanych dla wcześniejszych wersji systemu Mac OS
- łatwość adaptacji, kompilacji i wykorzystania oprogramowania stworzonego dla systemów uniksowych, zwłaszcza typu BSD
- Quartz Extreme, format PDF jako podstawa interfejsu graficznego
- ColorSync, przemysłowej jakości system zarządzania kolorem
- wydajny silnik OpenGL
- ma wbudowane zaawansowane narzędzia sieciowe

Na konferencji WWDC (WorldWide Developer Conference) 6 czerwca 2005 prezes Apple, Steve Jobs oświadczył, że Mac OS X był tworzony z myślą o procesorach firmy Intel i kompatybilnych. Zapowiedział przejście w ciągu dwóch lat całej linii komputerów Macintosh (a co za tym idzie - systemu operacyjnego i oprogramowania) na procesory Intela. Ma w tym pomóc innowacyjne oprogramowanie umożliwiające uruchamianie oprogramowania przewidzianego na procesory PowerPC na procesorach klasy x86. Jako przyczyny przejścia wymieniane były problemy firmy IBM z wyprodukowaniem energooszczędnych, szybkich procesorów do komputerów przenośnych.

MacOS X – wersje

- MacOS X 10.4 Tiger (2005r.)
 - pierwsza wersja systemu na platformę PC
- MacOS X 10.5 Leopard (koniec 2007r.)
 - polepszono interfejsy użytkownika nie pogarszając tym samym znacząco wydajności systemu
 - **Quick Look** – wyświetlanie, odtwarzanie i czytanie plików bez ich otwierania
 - **Time Machine** – narzędzie do tworzenia kopii zapasowych
 - **Kontrola rodzicielska** – zarządzanie, monitorowanie i kontrolowanie czasu, jaki dzieci spędzają przy komputerze, strony, które odwiedzają, a także osoby, z którymi prowadzą rozmowy
 - **Spaces** – tworzenie własnych przestrzeni roboczych w celu lepszej organizacji pracy
 - **Boot Camp** – uruchamianie systemu Windows XP/Vista na komputerach Mac

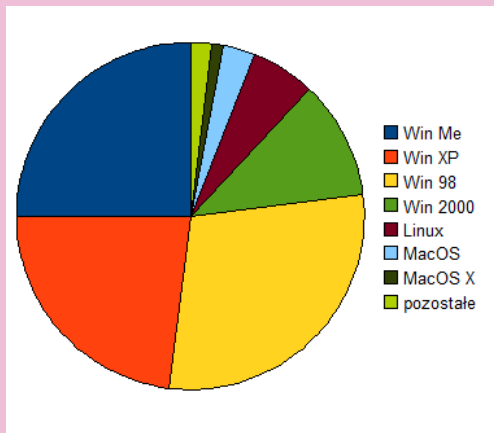
MacOS X – wersje, c.d.

- MacOS X 10.6 Snow Leopard (sierpień 2009)
 - rezygnacja ze wsparcia procesorów PowerPC
 - **Grand Central Dispatch** – technologia zwiększająca wsparcie dla aplikacji 64 bitowych, łatwe wykorzystanie wielordzeniowych procesorów przez programistów
 - **OpenCL** – Open Computing Language, wsparcie dla aplikacji wieloplatformowych składających się z różnego rodzaju jednostek obliczeniowych (CPU, GPU)
 - pełne wsparcie Microsoft Exchange 2007
- MacOS X 10.7 Lion (lato 2011)
 - sterowanie gestami
 - połączenie funkcjonalności MacOS X z iPada

Mac OS – pulpit

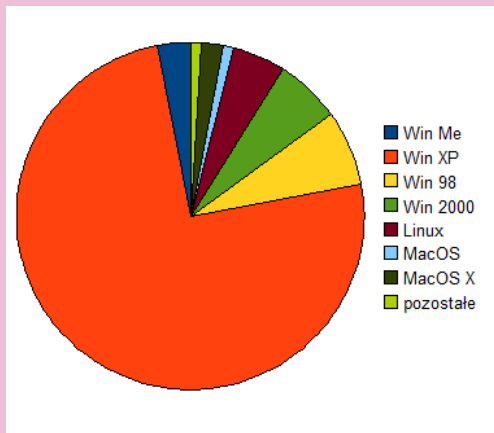


Rynek systemów operacyjnych – 2002



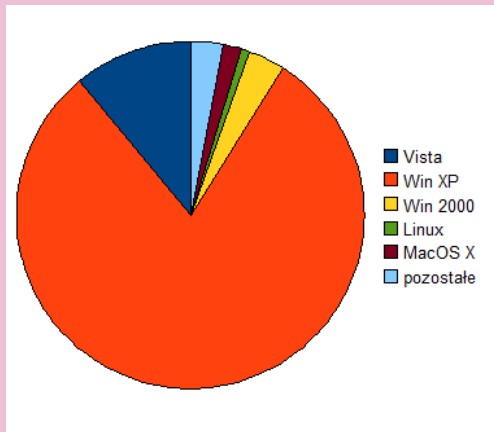
źródło: Bristol University

Rynek systemów operacyjnych – 2004



źródło: Bristol University

Rynek systemów operacyjnych – 2007

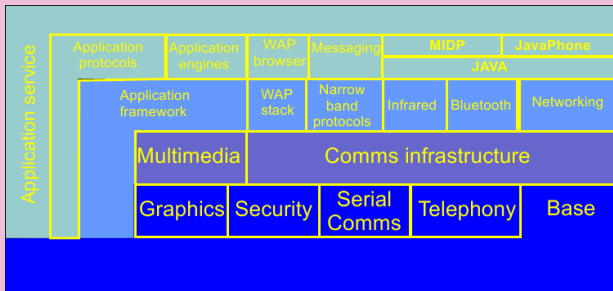


źródło: Geminus SA, użytkownicy internetu

Symbian OS

- Symbian jest złożonym systemem operacyjnym dedykowanym na platformę telefonów komórkowych
- Zbudowany jest z tzw. podsystemów
 - podsystem nie grupuje bibliotek czy plików źródłowych
 - w ramach podsystemu można wyróżnić tzw. funkcjonalności powiązane ze sobą

Struktura systemu



- **Base** – zawiera klasy do przechwytywania ciągów znaków, tablic, przechwytywania błędów i prostych typów danych. Zawiera także API dla zwykłych programistów, którzy potrzebują wykonywać operacje na wątkach, procesach i pamięci
- **Security** – zawiera komponenty, które dostarczają algorytmów kryptograficznych, zarządzania certyfikatami oraz aplikacji instalacyjnych zawierających cechy bezpieczeństwa
- **Graphics** – zawiera API do rysowania na szczególnych typach urządzeń takich jak ekrany i drukarki oraz przechwytywania czcionek i bitmap
- **Application framework** – podsystem zawiera definicje dotyczące struktury aplikacji oraz ich prostego interfejsu użytkownika. Aplikacja ma tutaj inne znaczenie niż program. Aplikacja traktowana jest jako program łącznie z interfejsem użytkownika

- **Engines and utilities** – zapewnia dostęp do danych aplikacji
- **Communications** – podsystem zawiera API związane z technologiami komunikacyjnymi
 - połączenia na podczerwień **IrDa**, **Bluetooth**, połączenie poprzez port szeregowy
 - **Telephony** zapewnia dostęp do funkcjonalności telefonu i określa zasady pisania sterowników dla określonego sprzętu
 - **Narrow bands Protocols** – zapewnia przechwytywanie wiadomości SMS
 - **Messaging** – udostępnia zasady dla wielo-protokołowego przesyłania wiadomości
 - **WAP Stack** – umożliwia dostęp do stosu protokołu WAP
 - **WAP Browser** – zapewnia pomoc dla WAP Push

Właściwości systemu

Przenaszalność

- Symbian jest systemem operacyjnym zbudowanym modularnie
- Istnieje możliwość dodawania, usuwania oraz modyfikowania komponentów
- Symbian zbudowany dla jednego typu telefonu może być, ale nie musi, kompatybilny z innym typem telefonu
- Znaczna część komponentów jest wspólna dla wielu typów telefonów komórkowych

Interfejs użytkownika

- Telefony dedykowane dla różnych grup użytkowników wymagają zupełnie innych mechanizmów na wprowadzanie czy wyprowadzanie informacji
- Symbian zarządza wymaganiami użytkowników oddzielając warstwę interfejsu użytkownika od systemu operacyjnego
- System udostępnia szkielet struktury i usług, zaś producent telefonu dostarcza prawidłowy interfejs użytkownika dla swojego telefonu

Wspomaganie standardów

- Symbian implementuje szeroką gamę standardów, głównie komunikacyjnych
- Bluetooth, CDMA, CompactFlash, Fax, GPRS, GSM, HSCSD, HTTP, IMAP4, IPv4, IPv6, IrDA, IrTranP, MIDP, MMS, MultiMediaCard, POP3, RS232, SMS, SMTP, SyncML, TCP/IP, Unicode, USB, WAP

Języki programowania

- C++ – Symbian jest napisany w C++, czyli jest to naturalny język do implementacji nowych programów
- Java – główny język programowania jako alternatywa do C++
- Assembler – używany rzadko, w sytuacjach kiedy wydajność jest priorytetowa
- JavaScript – obsługa stron HTML po stronie klienta
- WMLScript – język skryptowy do obsługi WAP

Rynek mobilnych systemów operacyjnych – 2008

