

# STRUKTURY SYSTEMÓW OPERACYJNYCH





# Agenda

- Komponenty systemu
- Usługi systemu operacyjnego
- Wywołania systemowe (ang. system calls)
- Programy systemowe
- Struktura systemu
- Maszyny wirtualne
- Projektowanie i implementacja systemu
- Generowanie systemu



# Podstawowe komponenty systemu

- Zarządzanie procesami
- Zarządzanie pamięcią główną
- Zarządzanie plikami
- System zarządzania operacjami wejścia/wyjścia
- Zarządzanie drugoplanowe
- Praca sieciowa
- System ochrony
- System interpretacji poleceń



# Zarządzanie procesami

- Proces jest to wykonujący się program. Procesowi do zrealizowania jego zadań niezbędne są pewne zasoby, włączając w to czas CPU, pamięć, pliki i urządzenia I/O.
- System operacyjny jest odpowiedzialny za następujące czynności związane z zarządzaniem procesami:
  - Tworzenie i usuwanie procesów
  - Zawieszanie i wznowianie procesów
  - Dostarcza mechanizmów do:
    - Synchronizacji procesów
    - Komunikacji pomiędzy procesami (miedzyprocesowej)



# Zarządzanie pamięcią główna

- Pamięć jest dużą tablicą słów lub bajtów; każdy element tablicy posiada dokładnie określony adres. Jest to repozytorium z szybko dostępnymi danymi, współdzielonymi przez CPU i urządzenia I/O.
- Pamięć główna jest ulotnym nośnikiem danych. W momencie wystąpienia awarii, wyłączenia systemu dane są tracone.
- System operacyjny jest odpowiedzialny za następujące czynności związane z zarządzaniem pamięcią:
  - Przechowuje informacje o tym, która część pamięci jest aktualnie zajęta i przez kogo.
  - Decyduje, który z procesów załadować w momencie, gdy pewien obszar pamięci staje się dostępny.
  - W razie potrzeby, przydziela i zwalnia określony obszar pamięci.



# Zarządzanie plikami

- Plik jest zbiorem powiazanych ze soba informacji, definiowanej przez twórcę. Pliki zawieraja zwykle programy (zarówno ich postac źródłowa, jak tez postac skompilowana) oraz dane.
- System operacyjny jest odpowiedzialny nastepujace czynnosci zwiazane z zarzadzaniem plikami:
  - Tworzenie i usuwanie plików.
  - Tworzenie i usuwanie katalogów.
  - Dostarcza podstawowych operacji wykonywanych na plikach i katalogach.
  - Odwzorowuje pliki w drugorzędne nośniki danych.
  - Kopie plików na stabilnych (nieuolotnych) nośnikach danych.



# System zarządzania I/O

- System I/O składa się z:
  - Systemowego, szybkiego bufora przechowywania informacji (ang. ang. buffer-caching system)
  - Ogólnego interfejsu do sterownika urządzenia
  - Sterownika określonego urządzenia.





## Zarządzanie pamięcią drugorzędna (podrzedna)

- Ponieważ pamięć główna (pamięć nadrzędna) jest ulotna i zbyt mała do ciągłego przechowywania wszystkich danych i programów, stąd system komputerowy musi dostarczyć pamięć drugorzędna (podrzedna), umożliwiającą tworzenie kopii pamięci głównej.
- Współczesne systemy komputerowe używają dysków jako głównego medium do przechowywania zarówno danych, jak też programów w trybie on-line.
- System operacyjny jest odpowiedzialny następujące czynności związane z zarządzaniem dyskiem:
  - Zarządzanie wolnymi obszarami
  - Przydzielanie obszaru do przechowywania informacji
  - Planowanie dostępu do dysku.





## Praca sieciowa (systemy rozproszone)

- System rozproszony jest zbiorem procesorów, które nie współdziela pamięci lub zegara. Każdy procesor ma własną lokalną pamięć lub zegar.
- Procesory w systemie połączone są poprzez sieć komunikacyjną.
- Komunikacja realizowana jest w oparciu o protokół komunikacyjny.
- System rozproszony zapewnia użytkownikowi możliwość dostępu do różnych zasobów systemu.
- Dostęp do wspólnych zasobów pozwala na:
  - Przyspieszenie obliczeń
  - Zwiększenie dostępności danych
  - Podniesienie poziomu niezawodności



# Ochrona systemu

- Ochrona odnosi się do mechanizmu kontrolowania dostępu programów, procesów, lub użytkowników zarówno do zasobów systemu, jak też użytkownika.
- Mechanizm ochrony musi:
  - Rozróżniać pomiędzy autoryzowanym i nieautoryzowanym użyciem zasobu.
  - Określić jakiego typu kontrola powinna być zastosowana
  - Dostarczyć środków wymuszenia kontroli.



# System interpretacji poleceń

- Wiele poleceń jest przekazywanych do systemu operacyjnego za pomocą **instrukcji sterujących**, które dotyczą:
  - Tworzenia i zarządzania procesami
  - Operacji I/O
  - Zarządzania drugorzędnymi nośnikami danych
  - Zarządzaniu pamięcią główną
  - Dostępu do systemu plików
  - Ochrony
  - Pracy w sieci



## System interpretacji polecen - c.d.

- Program, który czyta i interpretuje instrukcje sterujące nazywany jest różnie, np:
  - Interpreter wierszy polecen
  - shell (w UNIX)
- Jego zadanie jest proste – ma pobrać i wykonać kolejną instrukcję polecenia..



# Usługi systemu operacyjnego

- Wykonanie programu – zdolność systemu do załadowania programu do pamięci i uruchomienia go.
- Operacje I/O – ponieważ program użytkownika nie może wykonywać operacji I/O bezpośrednio, to środki do realizacji tych czynności musi mieć system operacyjny.
- Manipulowanie systemem plików – zdolność programu do czytania, zapisywania, tworzenia i usuwania plików.
- Komunikacja – wymiana informacji pomiędzy procesami wykonującymi się albo na tym samym komputerze albo na różnych systemach, połączonych ze sobą w sieć. Implementowana za pomocą **pamięci dzielonej** lub przy użyciu **przekazywania komunikatów**.
- Wykrywanie błędów – umożliwia relacje poprawnych obliczeń poprzez wykrywanie błędów w CPU i sprzeczności pamięci, w urządzeniach I/O lub w programach użytkownika.



## Dodatkowe funkcje systemu operacyjnego

- Dodatkowe funkcje nie istnieją po to, aby służyć pomocą użytkownikowi, ale raczej w celu zwiększenia efektywności i pewności operacji realizowanych przez system.
  - Alokacja zasobów – przydzielanie zasobów wielu użytkownikom lub wielu zadaniom pracującym w tym samym czasie..
  - Rozliczanie – przechowywanie informacji o tym, którzy użytkownicy i w jakim stopniu korzystają z poszczególnych zasobów w celu wystawienia rachunku lub po prostu w celu gromadzenia informacji w celach statystycznych.
  - Ochrona – stworzenie pewności, że wszystkie dostępy do zasobów systemu są kontrolowane.



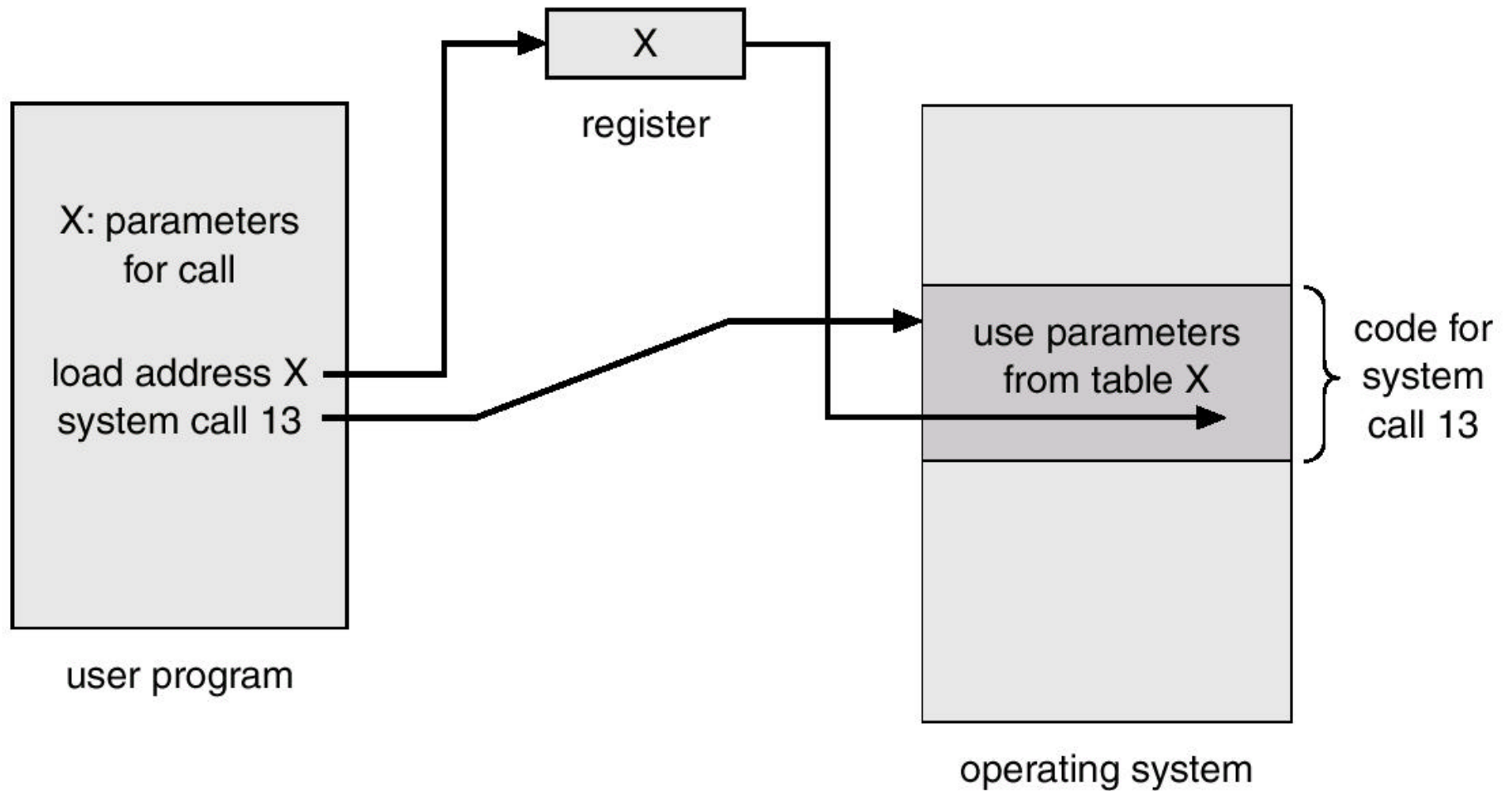
# Funkcje systemowe (wywołania systemowe)

- Funkcje systemowe dostarczają interfejsu między wykonywanym programem a systemem operacyjnym.
  - Zwykle dostępne w postaci instrukcji języka assembler.
  - Definiowane są języki zastępujące assembler dla potrzeb tzw. programowania, pozwalające na bezpośrednie wywołania funkcji systemowych np.. Z poziomu języka C, C++.
- Istnieją trzy podstawowe metody przekazywania parametrów pomiędzy wykonywanym programem a systemem operacyjnym.
  - Umieszczanie parametrów w rejestrach procesora.
  - Przechowywanie parametrów w tabeli w pamięci; adres tej tabeli jest przekazywany jako parametr w rejestrze procesora..
  - Włożenie (przechowywanie) parametrów na stosie przez program i zdejmowanie ich przez system operacyjny.





## Przekazywanie parametrów za pomocą tabeli





# Typy funkcji systemowych

- Kontrolowanie procesów
- Zarządzanie plikami
- Zarządzanie urządzeniami
- Utrzymywanie informacji
- Komunikacja

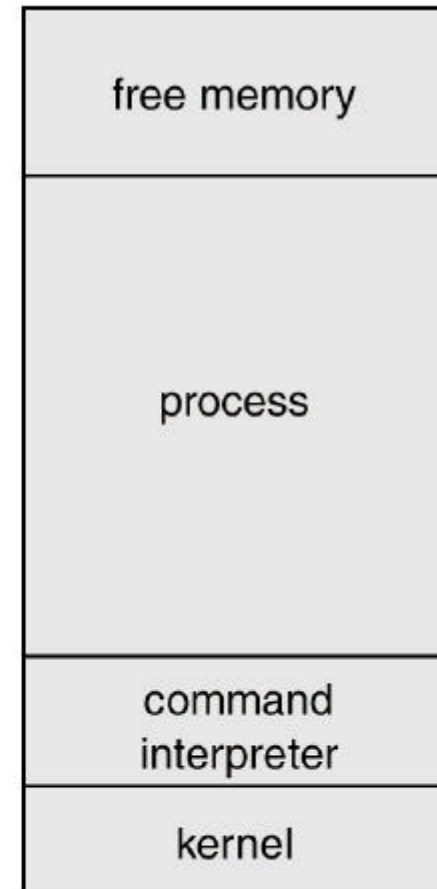


# Wykonanie programu w MS-DOS



(a)

W momencie startu systemu

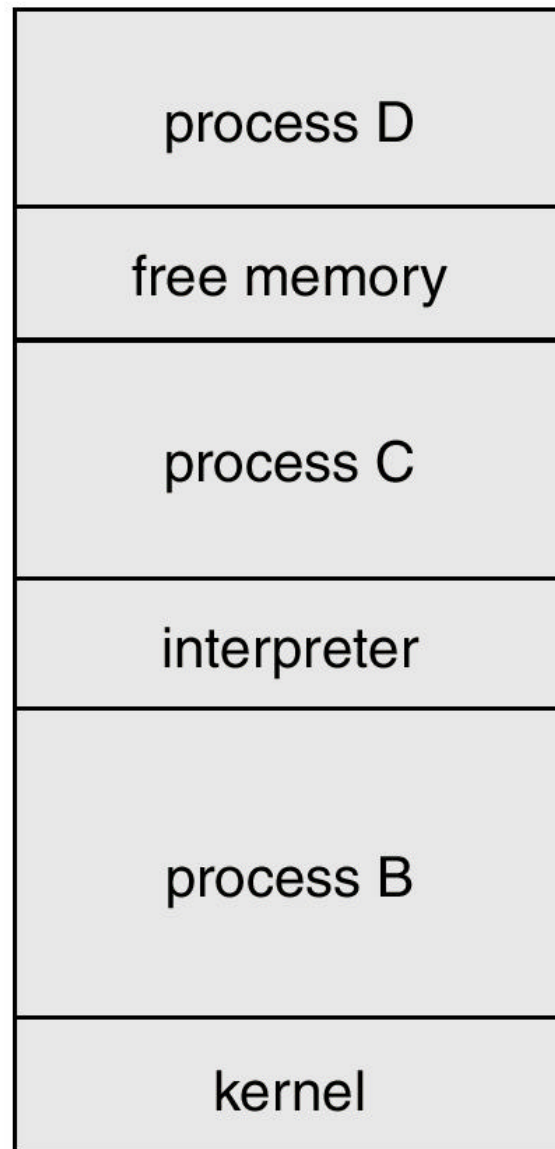


(b)

Uruchomienie programu



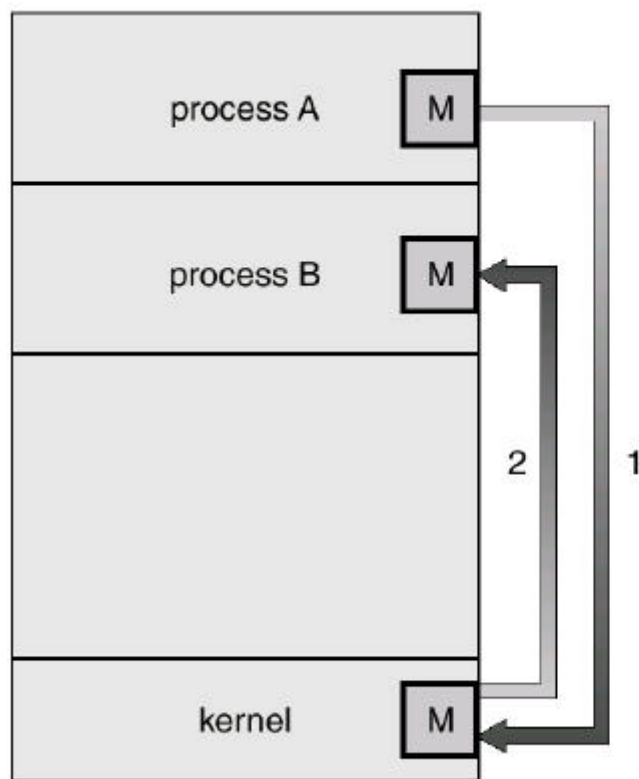
# Wykonywanie wielu programów w systemie UNIX



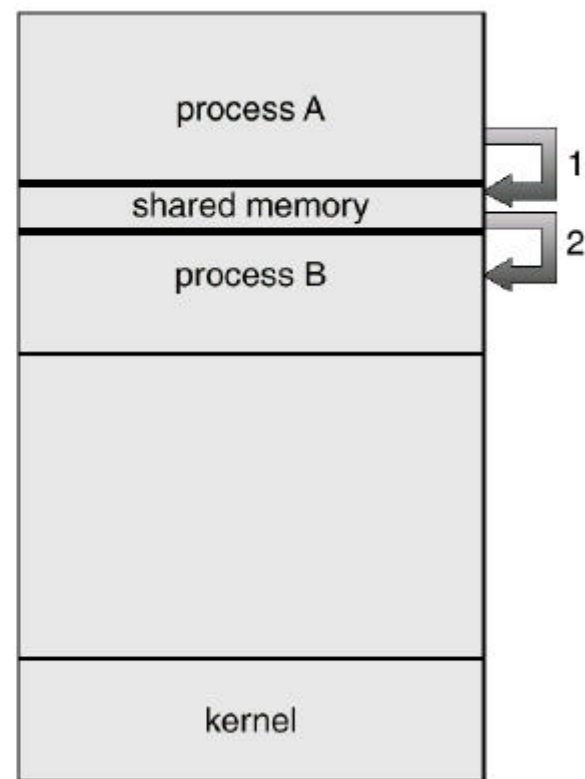


# Modele komunikacji

- Komunikacja może być realizowana za pośrednictwem przesyłania komunikatów lub pamięci dzielonej.



Przesyłanie komunikatów



Pamięć dzielona



# Programy systemowe

- Programy systemowe tworzą wygodne środowisko rozwijania i wykonywania programów. Można je podzielić na programy:
  - Manipulowanie plikami
  - Informowanie o statusie
  - Modyfikacja pliku
  - Wspierające języki programowania
  - Ładowanie i wykonywanie programów
  - Komunikacja
  - Programy użytkowe
- Większość użytkowników postrzega systemy operacyjne poprzez programy systemowe, nie zaś poprzez funkcje systemowe.



# Typy struktur systemów operacyjnych

- **Prosta struktura** (system monolityczny) - program złożony z wielu procedur, każda może wołać każdą, ograniczona strukturalizacja (np. OS/360, MS-DOS, Unix).
  - makrojadro - wszystkie funkcje SO umieszczone w jądrze i wykonywane w trybie uprzywilejowanym (np. OS/360)
  - mikrojadro - część funkcji przeniesiona z jądra na poziom użytkownika (np. Unix: jądro i programy systemowe)
- **Struktura warstwowa** - każda warstwa spełnia funkcje zależne tylko od warstwy poniżej (T.H.E., RC-4000, VMS); w Multicsie odmiana struktury warstwowej - zespół koncentrycznych pierścieni. Makrojadro – wszystkie warstwy są uprzywilejowane.
- **Struktura mikrojadra** - mikrojadro obejmuje jedynie tworzenie procesów, komunikację międzyprocesową, mechanizmy (ale nie strategię), zarządzanie pamięcią, procesorem i urządzeniami (na najniższym poziomie), np. Mach.



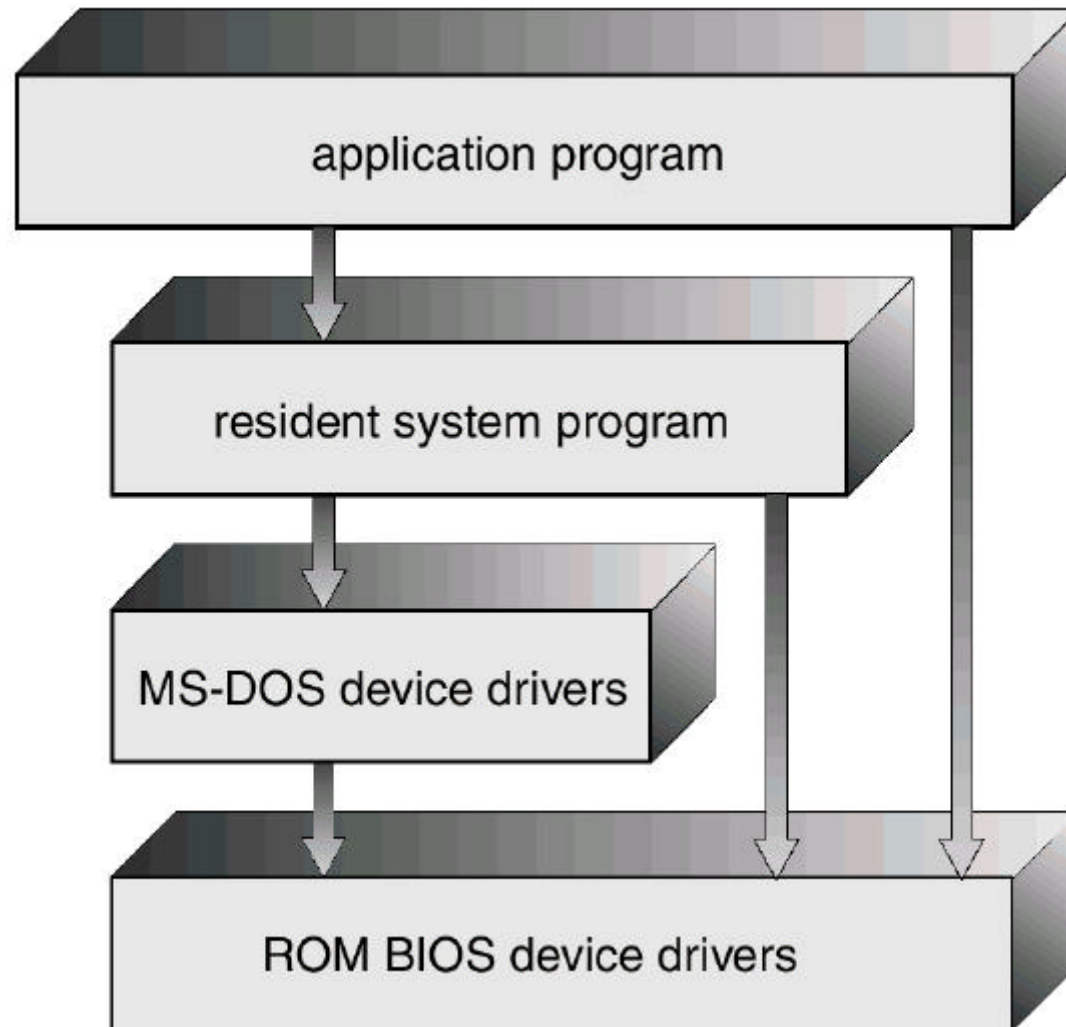


# Struktura systemu MS-DOS

- MS-DOS – system operacyjny pisany z myślą o zapewnieniu jak największej funkcjonalności przy jak najmniejszej wielkości (przy oszczędności miejsca).
  - Brak podziału na moduły
  - Mimo, że w systemie MS-DOS można wyodrębnić pewną strukturę, to jego interfejsy i poziomy funkcjonalne nie są wyraźnie wydzielone.



# Warstwowa struktura MS-DOS





# Struktura systemu UNIX

- UNIX – ograniczony przez funkcjonalność sprzętu oryginalny system operacyjny UNIX był przykładem ograniczonej strukturalizacji. System operacyjny UNIX składa się z dwóch oddzielnych części:
  - Programów systemowych
  - Jądra
    - Zawiera wszystko poniżej interfejsu do funkcji systemowych i powyżej sprzętu.
    - Zarządza systemem plików, planuje przydział CPU, zarządza pamięcią i wykonuje inne czynności systemu operacyjnego; wiele funkcji zebranych na jednym poziomie.



(the users)		
shells and commands compilers and interpreters system libraries		
<i>system-call interface to the kernel</i>		
signals terminal handling character I/O system terminal drivers	file system swapping block I/O system disk and tape drivers	CPU scheduling page replacement demand paging virtual memory
<i>kernel interface to the hardware</i>		
terminal controllers terminals	device controllers disks and tapes	memory controllers physical memory

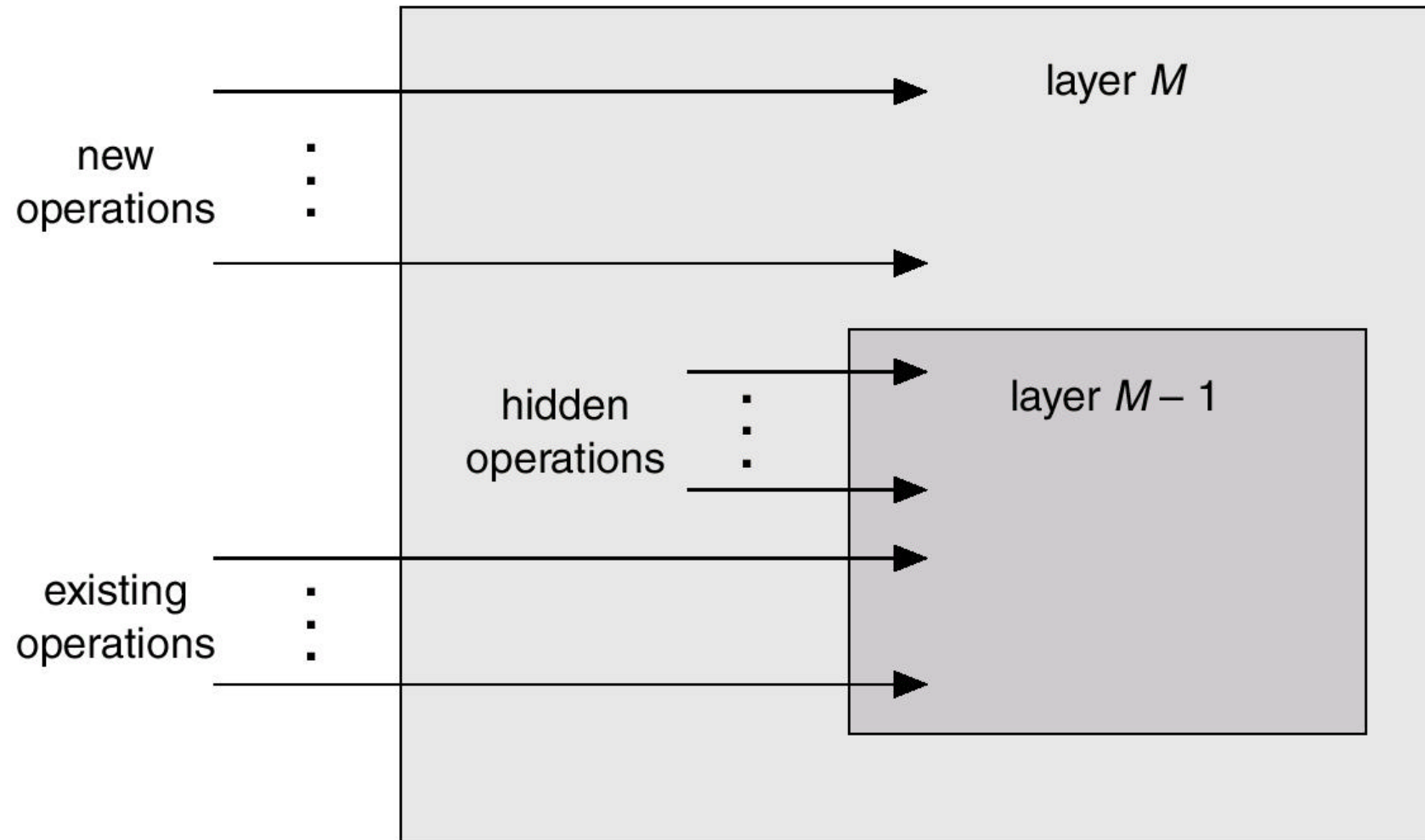


## Podajcie warstwowe

- System operacyjny jest dzielony na warstwy (poziomy), przy czym każda następna warstwa jest zbudowana na szczycie niższych warstw. Najniższa warstwa (warstwa 0) stanowi sprzęt, najwyższa (warstwa N) jest interfejs z użytkownikiem.
- Przy zachowaniu modularności warstwy są wybrane w ten sposób, że każda z nich używa funkcji (operacji) i korzysta z usług tylko niżej położonych warstw.

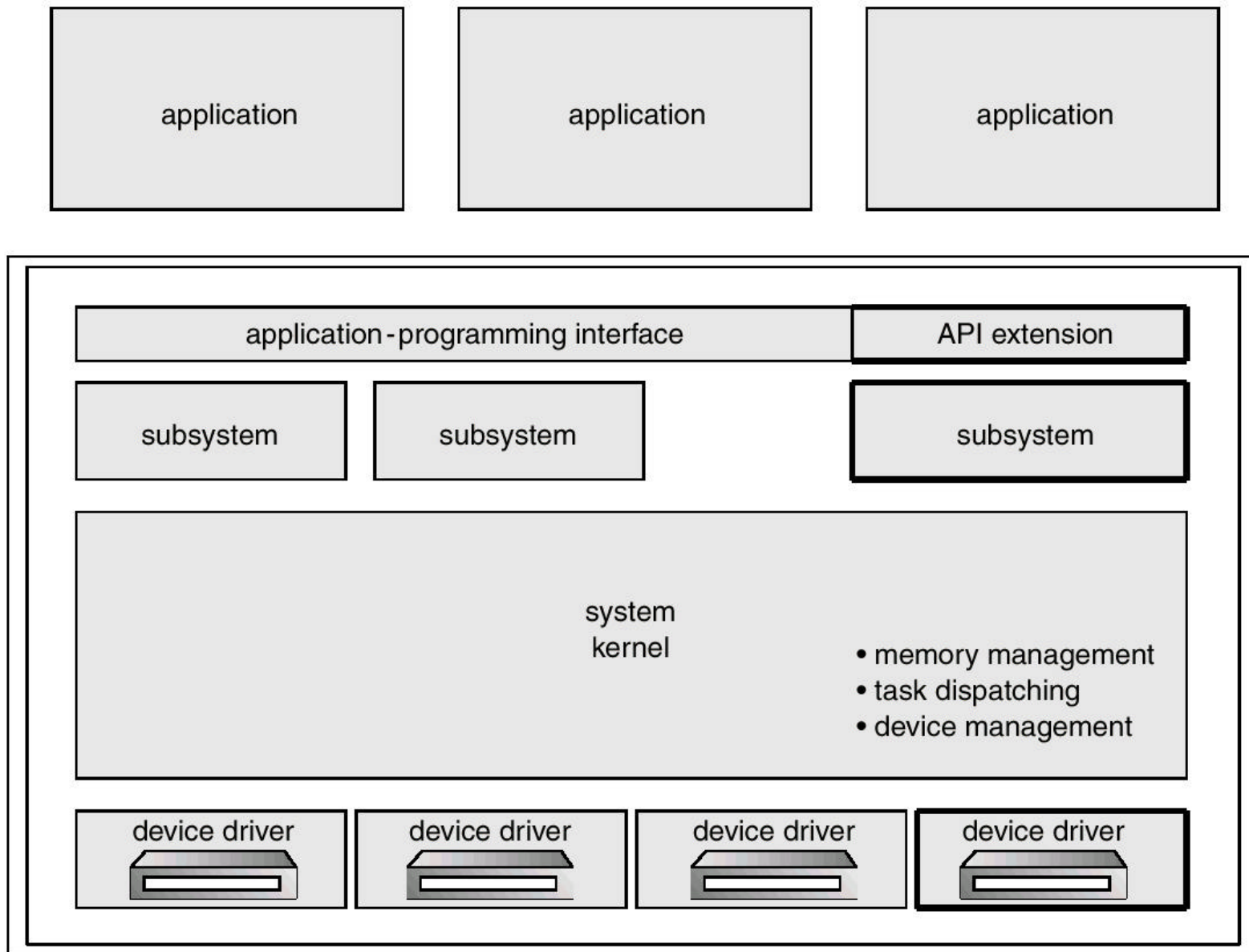


# Warstwy systemu operacyjnego





# Warstwowa struktura OS/2





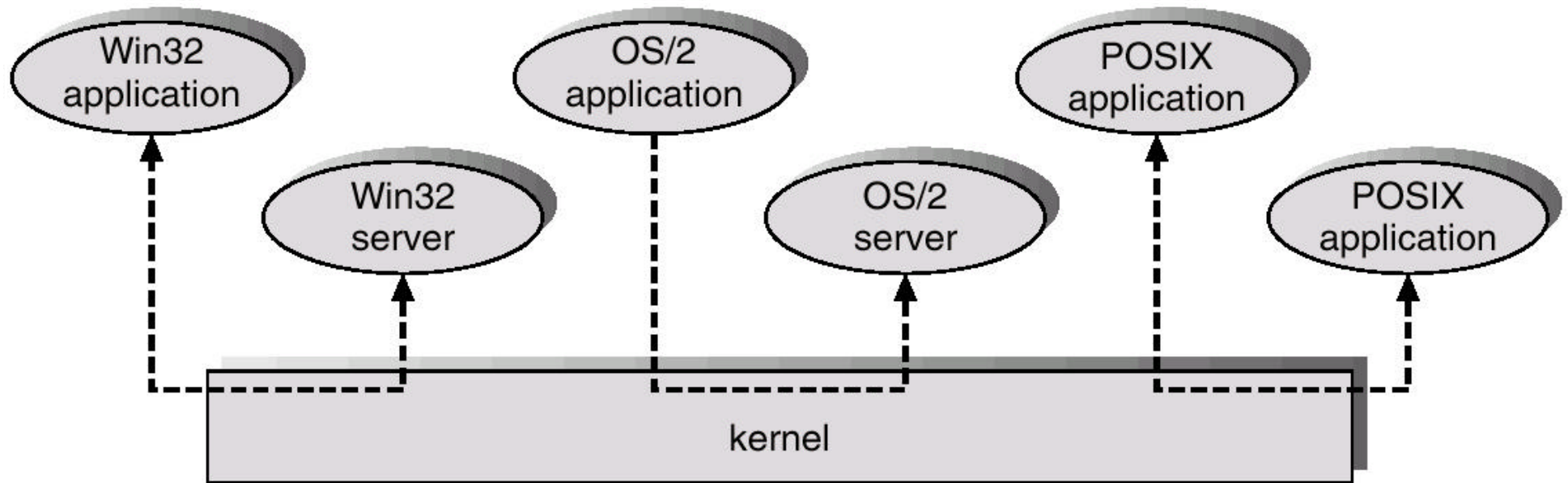


## Systemowa struktura mikrojadra

- Przemieszcza się tak jak to tylko jest możliwe z przestrzeni jądra do przestrzeni *użytkownika*.
- Komunikacja odbywa się pomiędzy modułami użytkownika przy zastosowaniu przesyłania wiadomości.
- Korzysci:
  - prościej można rozbudowywać mikrojądro
  - prościej można przenieść system operacyjny do nowej architektury
  - Większa niezawodność (mniej kodu wykonywane jest w trybie jądra)
  - Większe bezpieczeństwo.

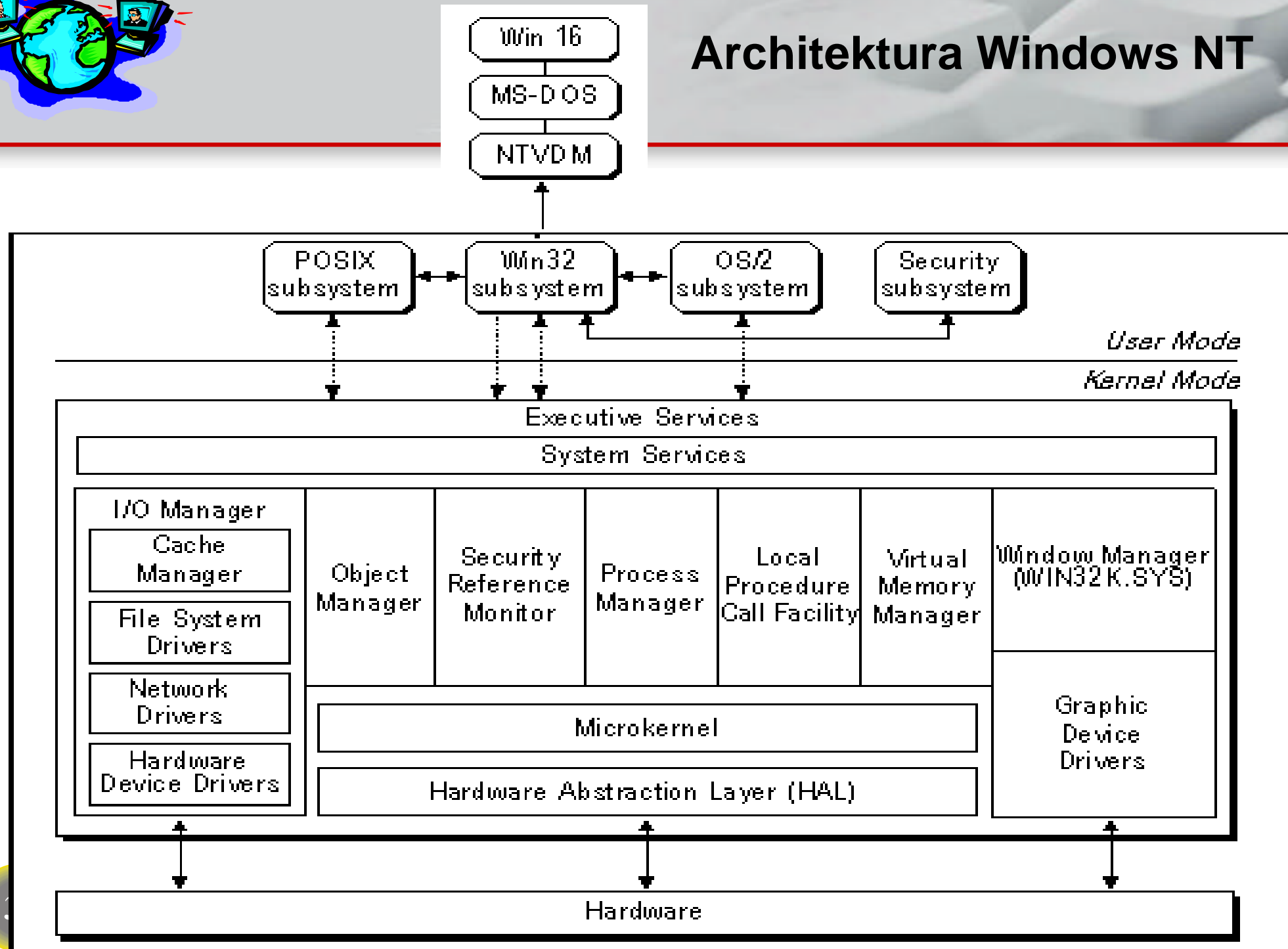


# Struktura klient-serwer sysetmu Windows NT





# Architektura Windows NT





## Maszyny wirtualne

- Logicznym rezultatem podejścia warstwowego jest maszyna wirtualna. Maszyna wirtualna traktuje sprzęt i jądro systemu operacyjnego tak, jak gdyby stanowiły one wszystkie wyłącznie sprzęt.
- Maszyna wirtualna dostarcza interfejs identyczny z podstawowym (bazowym) sprzętem.
- System operacyjny tworzy iluzję obecności wielu procesów, z których każdy wykonywany jest na własnym procesorze z własną (wirtualną) pamięcią.

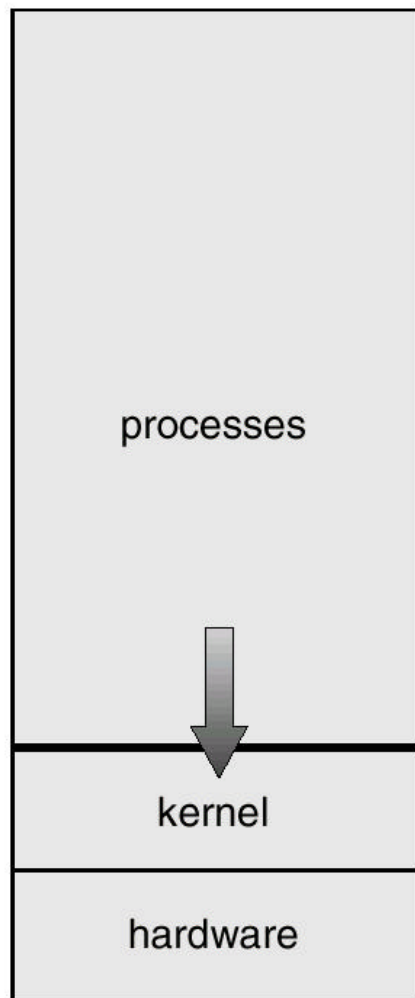


## Maszyny wirtualne (c.d.)

- Zasoby fizyczne komputera są dzielone tak, aby tworzyć wirtualną maszynę.
  - Planowanie przydziału CPU może tworzyć wrażenie, że użytkownicy posiadają własne procesory.
  - Spooling i system plikowy pozwalają utworzyć wirtualne dyski (minidyski) i wirtualne drukarki.
  - Zwykły terminal z podziałem czasu zachowuje się jak wirtualna maszyna konsoli operatora.

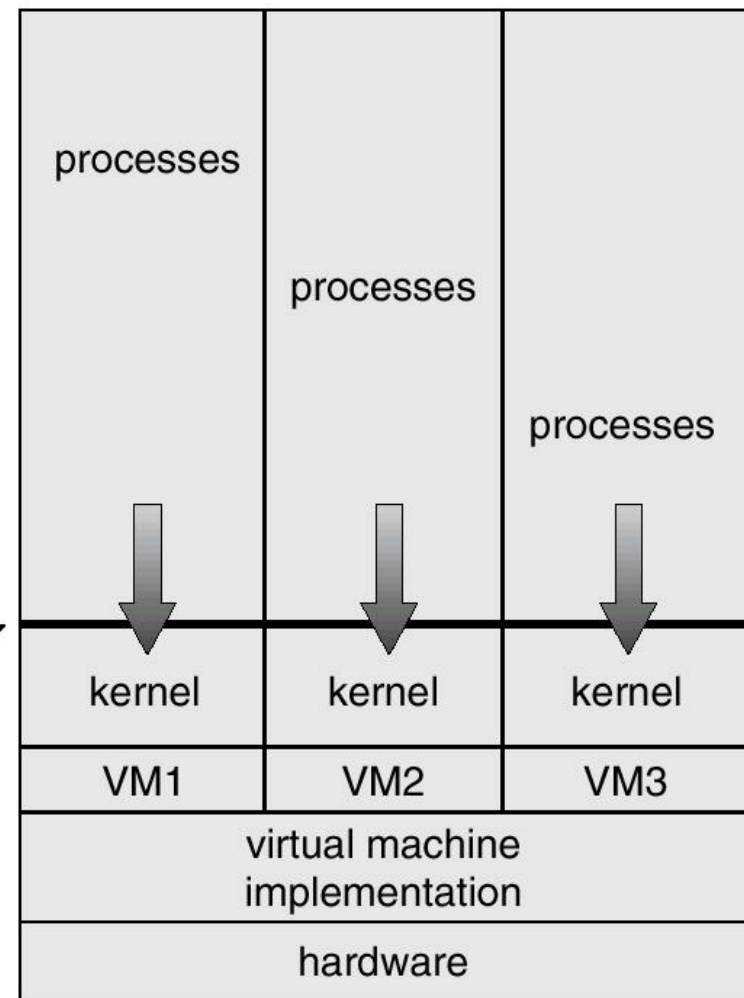


# Model systemów



Maszyna niewirtualna

programming interface



Maszyna wirtualna



## Zalety/wady maszyn wirtualnych

- Maszyna wirtualna zapewnia pełną ochronę zasobów systemu ponieważ każda maszyna wirtualna jest odizolowana od innej maszyny wirtualnej. Jednak izolacja nie zezwala na bezpośrednie współdzielenie zasobów.
- System maszyny wirtualnej stanowi wspierającą platformę do badań nad systemami operacyjnymi i ich rozwojem. Rozwój systemu realizowany jest na bazie maszyny wirtualnej, zamiast na bazie maszyny rzeczywistej i nie zakłóca normalnych operacji systemu.
- Koncepcja maszyny wirtualnej jest trudna do zaimplementowania. Zrealizowanie dokładnej kopii maszyny bazowej wymaga wielkiego wysiłku.



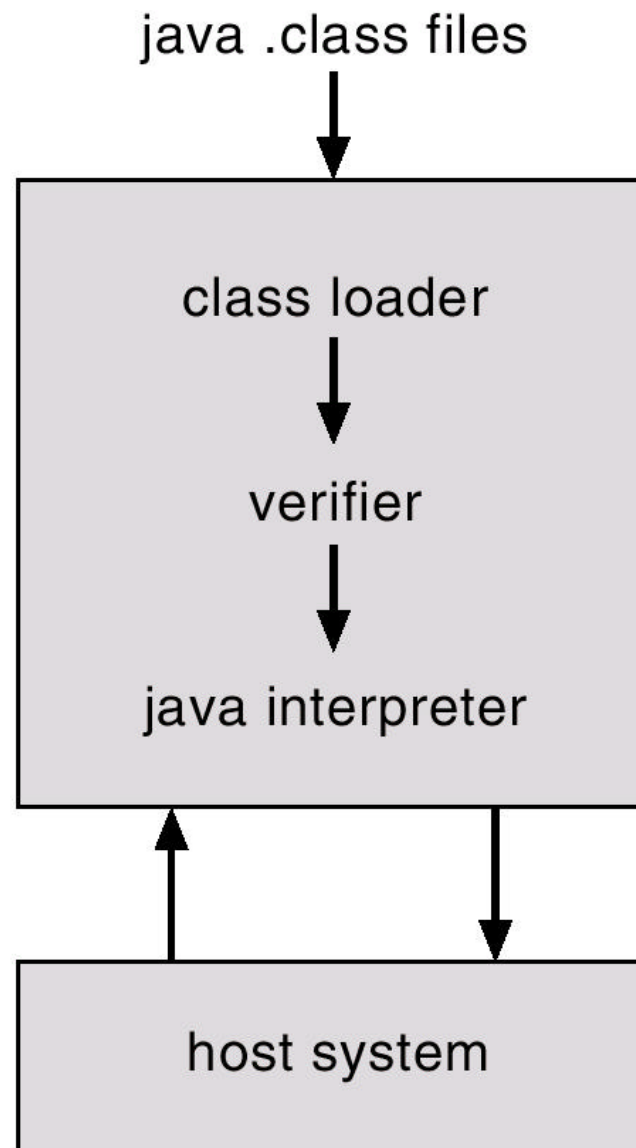


# Maszyna wirtualna Java

- Skompilowane programy napisane w języku Java zawierają kod bajtowy niezależny od platformy sprzętowej, wykonywany w środowisku Java Virtual Machine (JVM).
- JVM składa się z:
- Programu ładowania klas
- Weryfikatora klas
- Interpretera w czasie wykonywania
- Kompilatora typu *w odpowiednim momencie* (ang. Just-In-Time (JIT)) zwiększa wydajność maszyny.



# Maszyna wirtualna Java





## Zalozenia projetowe systemu operacyjnego

- Cele uzytkownika – system opeacyjny powinien byc wygodny w uzyciu, latwy do nauki, niezawodny, bezpieczny i szybki.
- Cele systemu – system operacyjny powinien byc latwy do zaprojektowania, zaimplementowania i utrzymania, a takze byc elestyczny, niezawodny, pozbawiony bledów i wydajny.



## Mechanizmy a polityka

- Mechanizmy określają jak czegoś dokonać, natomiast polityka decyduje o tym, co ma być zrobione.
- Oddzielenie polityki od mechanizmu jest bardzo ważną zasadą. Pozwala to na uzyskanie maksymalnej elastyczności w sytuacji, gdy decyzje polityczne zostaną później zmienione.



## Implementacja systemu

- Tradycyjnie pisane w języku assembler obecnie systemy operacyjne mogą być pisane w językach wysokiego poziomu.
- Kod pisany w języku wysokiego poziomu:
  - Może być pisany szybciej.
  - Jest bardziej zwarty.
  - Jest prostszy do zrozumienia i uzdatniania (ang. debugg).
- System operacyjny jest znacznie łatwiejszy do *przenoszenia* (instalowania na innym sprzęcie) jeśli jest napisany w języku wysokiego poziomu. Na przykład system MS-DOS napisano tylko w języku assembler mikroprocesora Intel, natomiast system Unix prawie w całości został napisany w języku C.



## Generowanie systemu (SYSGEN)

- Systemy operacyjne projektowane są tak, aby działały na pewnej klasie maszyn w rozmaitych instalacjach ze zmienną konfiguracją urządzeń zewnętrznych..
- System operacyjny musi podlegać konfigurowaniu lub generowaniu dla każdej specyficznej instalacji komputerowej. Proces ten nazywa się **generowaniem systemu**.
- Program SYSGEN gromadzi informacje dotyczące określonej konfiguracji systemu sprzętowego.
- *Booting* (wprowadzenie programu do pamięci komputera przy użyciu kilku instrukcji inicjujących) – uruchomienie komputera poprzez załadowanie jądra.
- *Program ładujący* (ang. *bootstrap program*) – kod przechowywany w pamięci ROM zdolny do zlokalizowania jądra, załadowania go do pamięci i uruchomienia go.





**DZIEKUJE PANSTWU**

**Jesli sa pytania, to z  
przyjemnoscia na nie  
odpowiemy**

