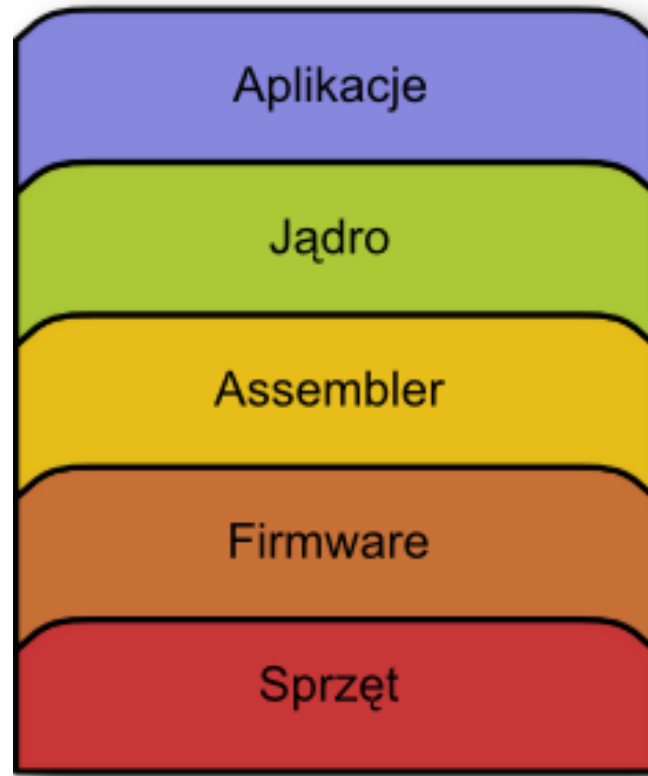


Systemy operacyjne

Struktura systemu operacyjnego

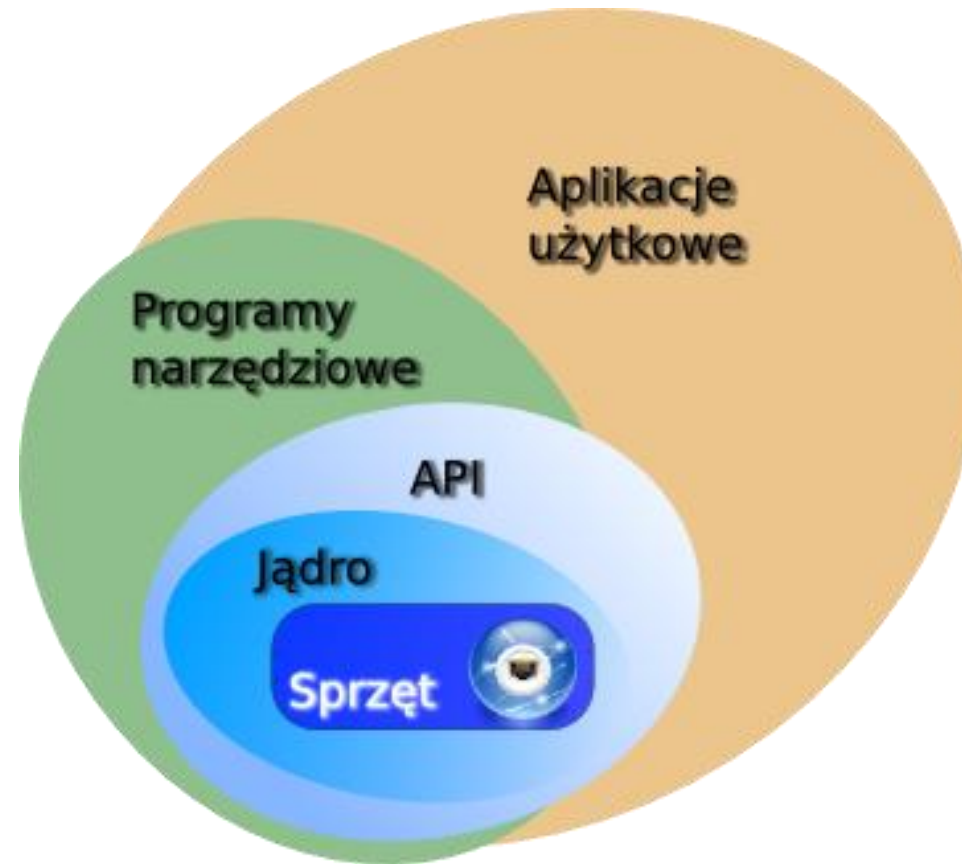
Systemy operacyjne

Schemat budowy systemu operacyjnego
model warstwowy



Systemy operacyjne

Schemat budowy systemu operacyjnego
części składowe



Systemy operacyjne

Większość systemów operacyjnych opiera się o koncepcję jądra, która w naturalny sposób wynika z modelu warstwowego.

W takim wypadku jądro jest po prostu nazwą nadaną najniższej warstwie w pełni implementowanej jako program wykonywany przez CPU.

Systemy operacyjne

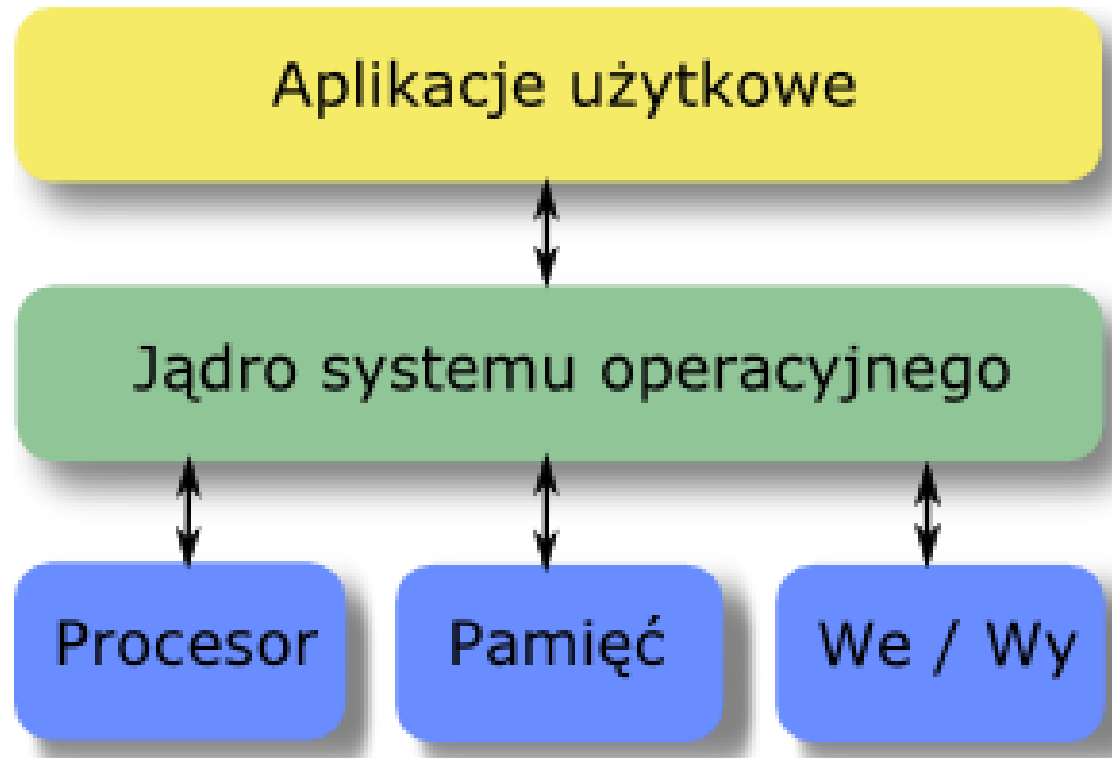
Najważniejsze zadania jądra systemowego:

- zarządzanie procesami
- zarządzanie pamięcią operacyjną
- obsługa systemu wejścia i wyjścia
- zarządzanie plikami i przestrzenią dyskową
 - uwierzytelnianie i ochrona
- implementacja* interfejsu poleceń

**Implementacja (wdrożenie, przystosowanie, realizacja, łac.ang. implementation) – w informatyce – proces przekształcania abstrakcyjnego opisu systemu lub programu na obiekt fizyczny: komputer lub działający program zapisany w konkretnym języku programowania; także obiekt fizyczny będący efektem takiego przekształcenia, np. implementacja systemu operacyjnego (wdrożenie systemu) lub kompilatora dla konkretnego typu komputera.*

Systemy operacyjne

Jądro monolityczne



Systemy operacyjne

Jądro monolityczne

Często stosowane w systemach Unix-owych. Wszystkie zadania są wykonywane przez jądro, będące jednym, dużym programem pracującym w pełni (wszystkie funkcje) w trybie jądra.

Zaletami tej techniki są: prostota, stabilność oraz łatwość komunikacji pomiędzy różnymi członami jądra (to przecież w tym wypadku jeden program!)

Systemy operacyjne

Jądro monolityczne

Dodatkowo, systemy z jądrem monolitycznym zazwyczaj działają szybciej (szczególnie w komputerach o małej liczbie CPU) niż ich główna alternatywa - mikrojądro.

Główna wada natomiast wynika z problemów w rozwijaniu takiego jądra (ze względu na jego wielkość) oraz dodawaniu do niego procedur obsługi nowych urządzeń peryferyjnych.

Dodanie obsługi nowego urządzenia wymaga kompilacji całego jądra, co zajmuje czas i jest trudne do wykonania przez nawet zaawansowanego użytkownika.

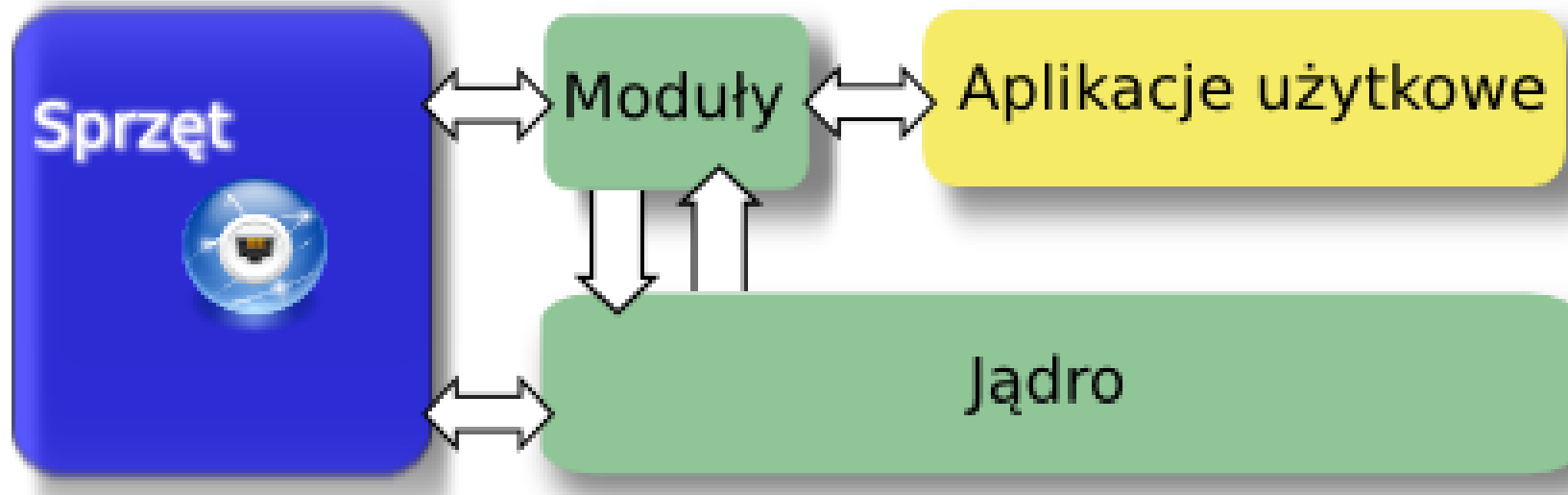
Systemy operacyjne

Jądro monolityczne

W czystej postaci jądro monolityczne występuje tylko w zastosowaniach niszowych, natomiast w dużej mierze takie rozwiązanie stosują systemy Linux, OpenBSD czy FreeBSD - jądra tych systemów umożliwiają ładowanie modułów, co jest sprzeczne z koncepcją jednego dużego programu, ale daje możliwość stosowania driver'ów, czyli dodawania obsługi nowego sprzętu bez konieczności ponownej kompilacji całego jądra.

Systemy operacyjne

Mikrojądro



Systemy operacyjne

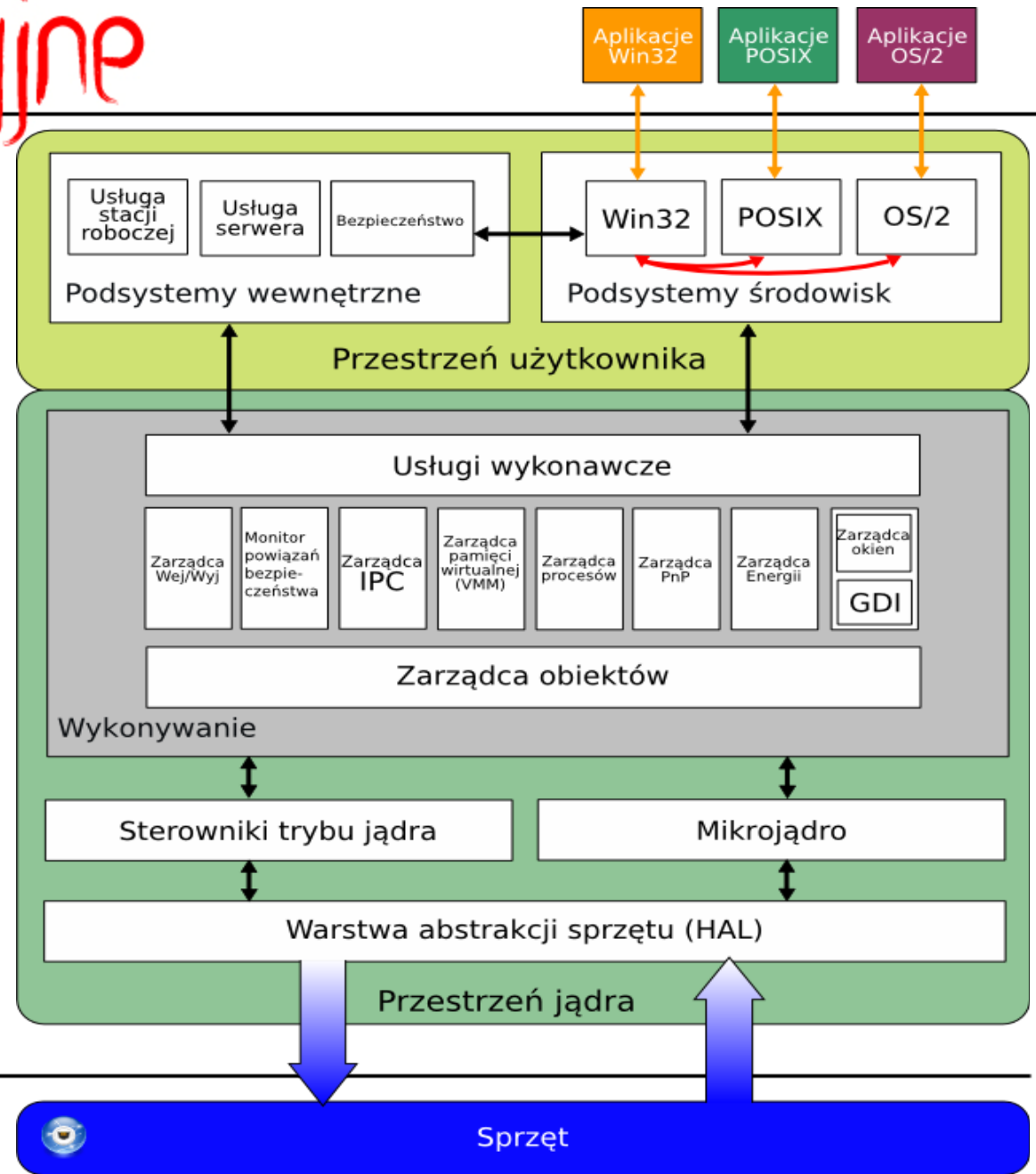
Mikrojądro

W tej technice z monolitycznego jądra zostaje tylko jego podstawowa część (odpowiedzialna za zarządzanie procesami oraz pamięcią). Części odpowiedzialne za bardziej wyrafinowane funkcje (m.in. obsługę urządzeń peryferyjnych) są wydzielone do funkcjonalnych bloków albo realizowane jako zwykłe procesy w trybie użytkownika. W większości przypadków takie bloki funkcjonalne mogą być ładowane i resetowane nie przerywając pracy systemu komputerowego, ponadto zwiększanie funkcjonalności jądra (i w konsekwencji - całego systemu) jest możliwe bez wprowadzania jakichkolwiek zmian w jego podstawowej części.

Główną wadą takiego podejścia jest wolniejsza praca systemu.

Systemy operacyjne

Jądro hybrydowe



Opracował: mgr Marek Kwiatkowski

Systemy operacyjne

Jądro hybrydowe

Jest to kompromis między architekturą jądra monolitycznego i mikrojądra.

W krytycznych usługach - np. stos sieci - usługi są na stałe wkompiłowane w główny kod jądra, inne usługi pozostają oddzielone od głównego jądra i działają jako serwery (w przestrzeni jądra).

Dzięki temu rozwiązaniu możliwe jest zachowanie wydajności jądra monolitycznego dla kluczowych usług.

Systemy operacyjne

Jądro hybrydowe

Wydzielenie takiego rodzaju jądra budzi kontrowersje wśród specjalistów. Wielu z nich uważa, iż w porównaniu do podobnego jądra monolitycznego, pojęcie to jest niczym innym jak dobrym marketingiem.

Pomysłem stojącym za tym nowym pseudo-rodzajem jądra jest struktura podobna do mikrojądra, ale zaimplementowana jak jądro monolityczne. W przeciwieństwie do mikrojądra, wszystkie (lub prawie wszystkie) usługi wykonywane są w przestrzeni jądra. Podobnie jak w jądrze monolitycznym, nie ma strat w wydajności wywołanych przepływem komunikatów mikrojądra. Jednakże, podobnie jak w jądrach monolitycznych, nie ma korzyści wynikających z umieszczenia usług w przestrzeni użytkownika.

Systemy operacyjne

Interpreter poleceń

Kolejnym elementem SO jest interpreter poleceń, który może być zaimplementowany w jądrze systemu lub przyjmować postać odrębnego programu np. interfejsu graficznego

Systemy operacyjne

W zależności jakie jądro jest zastosowane w systemie dany system operacyjny może cechować:

- Wielozadaniowość – wiele zadań, dzielenie czasu mikroprocesora
- Wielodostępność – wiele użytkowników może pracować w tym samym czasie
 - Wielowątkowość – wykonanie jednego procesu w ramach kilku wątków
 - Wielobieżność – kilka procesów ma dostęp do interfejsu jądra i przez to wszystkie mają dostęp do funkcji systemowych
 - Skalowalność – możliwość łatwej rozbudowy elementów systemu operacyjnego. Ważne jest żeby w miarę zwiększania objętości systemu nie spadała jego wydajność
- Wywłaszczanie – ta technika pozwala na wstrzymanie jednego procesu, aby umożliwić uruchomienie drugiego.