

---

# Sieciowe Systemy Operacyjne

Wprowadzenie do wykładu, podstawowe definicje, rola systemu operacyjnego	1
Procesy POSIX, zarządzanie procesami	2
Pliki, komunikacja przez pliki, blokowanie	1
Łącza nazwane i nienazwane	1
Kolejki komunikatów POSIX	1
Pamięć dzielona	1
Komunikacja międzyprocesowa, synchronizacja procesów, problem producenta konsumenta	1
Semaforey POSIX	1
Programowanie w interfejsie gniazd BSD	2
Protokoły TCP i UDP - właściwości, programowanie, aplikacje	1
Usługi sieciowe: telnet, FTP, ssh, DNS, dhcp, demon sieciowy inetd	1
Zdalne wywoływanie procedur RPC , standard XDR, aplikacje	1
Kolokwium zaliczeniowe	1
RAZEM	15

## Literatura:

[1] Abraham Silberschatz, Peter Galvin, Podstawy systemów operacyjnych, WNT Warszawa 2000.

[2] William Stallings, Systemy operacyjne Struktura i zasady budowy, PWN Warszawa 2006.

[3] K. Haviland, Dina Gray, Ben Salama, UNIX Programowanie systemowe, wyd. RM Warszawa 1999.

[4] John Fusco, Linux – niezbędny programista, Helion 2009.

[5] Douglas E. Comer, Sieci komputerowe i intersieci, WNT Warszawa 2000.

[6] M. Ben-Ari, Podstawy programowania współbieżnego i rozproszonego, WNT Warszawa 1996.

[7] Podręcznik języka C - <http://pl.wikibooks.org/wiki/C>

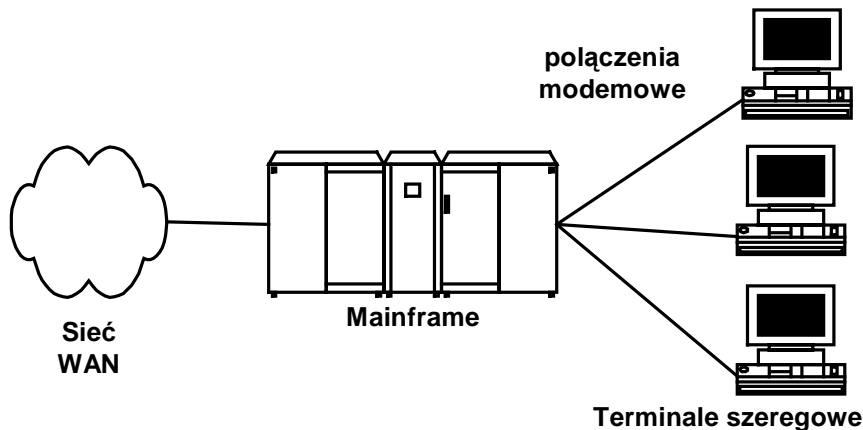
## 1. Etapy rozwoju systemów komputerowych

- System scentralizowany
- System sieciowy
- System rozproszony

### 1.1 System scentralizowany

Zastosowane duże komputery (mainframes )

Użytkownicy połączeni poprzez zdalne terminale znakowe



Rys. 1-1 Struktura systemu scentralizowanego

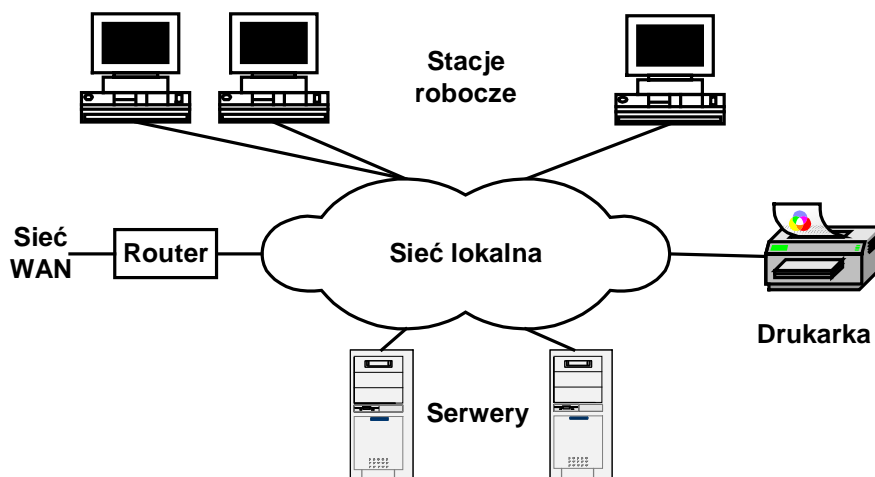
### 1.2 System sieciowy

System sieciowy składa się z wielu stacji roboczych i serwerów połączonych siecią. System sieciowy zawiera komponenty pozwalające na współdziałanie pomiędzy stacjami roboczymi a serwerami. Użytkownik jest świadomy że pracuje w sieci złożonej z wielu komputerów. Stosowana jest wspólna struktura komunikacji. Sieciowy system operacyjny tworzy środowisko w którym użytkownicy mają dostęp do wspólnych zasobów.

Pierwszy szeroko rozpowszechniony system sieciowy – Unix BSD

Zastosowane mechanizmy:

- Sieciowy system plików NFS (ang. *Network File System*)
- Zdalne wywoływanie procedur RPC (ang. *Remote Procedures Calls*)
- Sieciowe usługi informacyjne NIS (ang. *Network Information Services*)



Rys. 1-2 System sieciowy

Właściwości:

- Luźno powiązany sprzęt - różnego typu stacje robocze i serwery połączone siecią LAN.
- Różne systemy operacyjne - UNIX, Windows
- Obliczenia wykonywane przeważnie na maszynie lokalnej
- Pewien zbiór wspólnych usług i protokołów współpracy – wspólne serwery plików, poczta elektroniczna, TCP/IP

Zalety:

- Dzielenie zasobów
- Łatwa komunikacja między stacjami

Wady:

- Użytkownicy koncentrują coraz więcej zasobów w swoich komputerach które w większości pozostają niewykorzystane.
- Zasoby niewykorzystane nie mogą być udostępnione innym
- Istnieją zadania niewykonalne na indywidualnych stacjach roboczych

Przykłady sieciowych systemów operacyjnych:

Unix BSD, Linux, Novel Netware, Microsoft Windows 2000, XP, Solaris.

### 1.3 System rozproszony

Czynniki które umożliwiły powstanie systemów rozproszonych:

- Rozpowszechnienie tanich i wydajnych komputerów
- Rozwój sieci komputerowych

Naturalna jest tendencja aby dokonać ściślejszego połączenia komputerów co prowadzi do systemów rozproszonych.

#### **Główna idea systemów rozproszonych:**

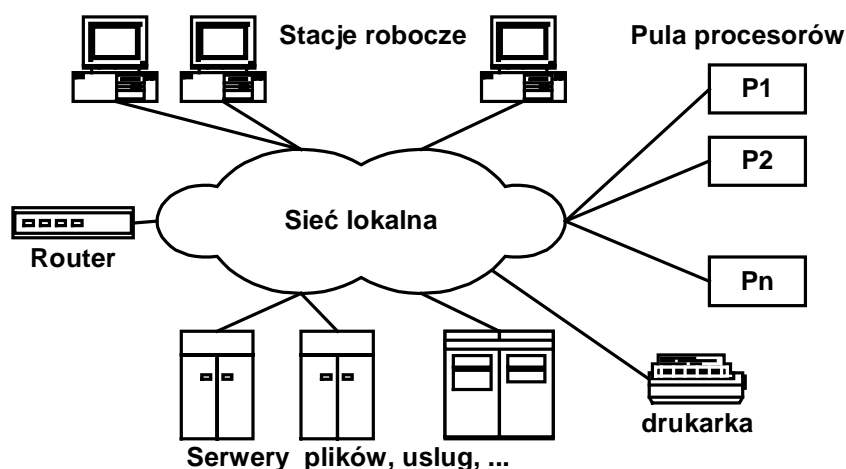
Integracja zasobów sprzętowych i programowych wielu komputerów za pomocą sieci w celu wzajemnego udostępnienia ich użytkownikom.

#### **Definicja:**

System rozproszony jest to układ niezależnych komputerów który sprawia na jego użytkownika wrażenie że jest jednym komputerem.

Własności:

- Komputery połączone mniej lub bardziej ściśle
- Na wszystkich komputerach wykonywany jednego typu system operacyjny
- Komputery mogą być różnego typu
- Jednolity mechanizm komunikacji pomiędzy procesami
- System plików wszędzie wygląda jednakowo



Rys. 1-3 Architektura sprzętowa systemu rozproszonego

Oprogramowanie systemu rozproszonego – rozproszony system operacyjny

Przykłady rozproszonych systemów operacyjnych:  
Amoeba, Mach, Chorus, DCE

## 2. Systemy Operacyjne

### 2.1 Funkcje systemu operacyjnego

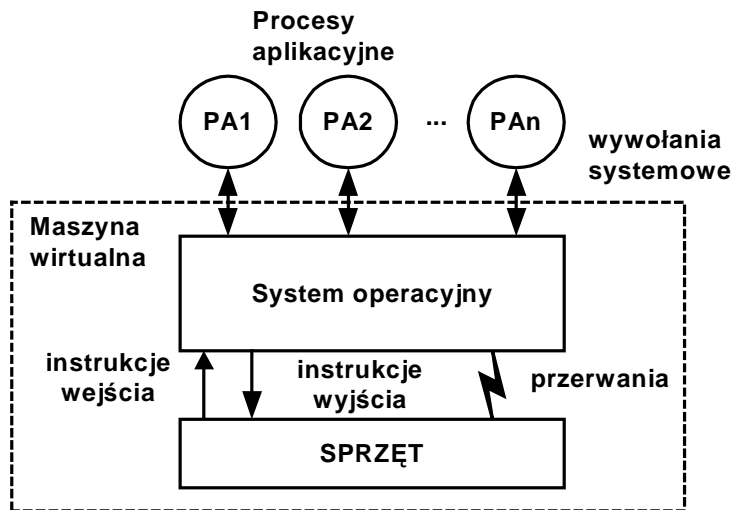
System operacyjny jest warstwą oprogramowania leżącą pomiędzy sprzętem a programami aplikacyjnymi. Najważniejsze funkcje systemu operacyjnego są następujące:

1. Implementacja współbieżności - procesów i wątków
  2. Zarządzanie urządzeniami wejścia wyjścia
  3. Implementacja pamięci wirtualnej
  4. Implementacja systemu plików
  5. Implementacja protokołów komunikacyjnych
  6. Implementacja interfejsu z użytkownikiem
  7. Zapewnienie bezpieczeństwa
- System operacyjny umożliwia efektywne wykorzystanie mocy obliczeniowej procesorów poprzez umożliwienie współbieżnego ich wykorzystania poprzez wielu użytkowników, wiele procesów i wątków.
  - Dostarcza narzędzi do bezpiecznego współdzielenia zasobów systemu pomiędzy użytkownikami i procesami.
  - Zwalnia użytkowników od uciążliwości programowania złożonych urządzeń wejścia - wyjścia.
  - Dostarcza abstrakcyjnego mechanizmu pamięci operacyjnej w postaci pamięci wirtualnej oraz abstrakcyjnej pamięci trwałej w postaci systemu plików.

Komunikacja pomiędzy programami aplikacyjnymi a systemem operacyjnym odbywa się poprzez wywołania systemowe (*ang. system calls*). Ma ono najczęściej postać funkcji lub procedury w której znaczenie parametrów i ich typy są dokładnie zdefiniowane i opisane. Zbiór wywołań systemowych tworzy interfejs do systemu operacyjnego API (*ang. Application Program Interface*).

System operacyjny komunikuje się ze sprzętem za pomocą instrukcji wejścia - wyjścia i reaguje na zgłaszane przez sprzęt przerwania.

System operacyjny pozwala ukryć przed programistą złożoność sprzętu. Komputer widziany jest przez programistę jako abstrakcyjny mechanizm zdolny wykonywać instrukcje języka programowania i zbiór wywołań systemowych. Mechanizm ten nazywa się maszyną wirtualną.



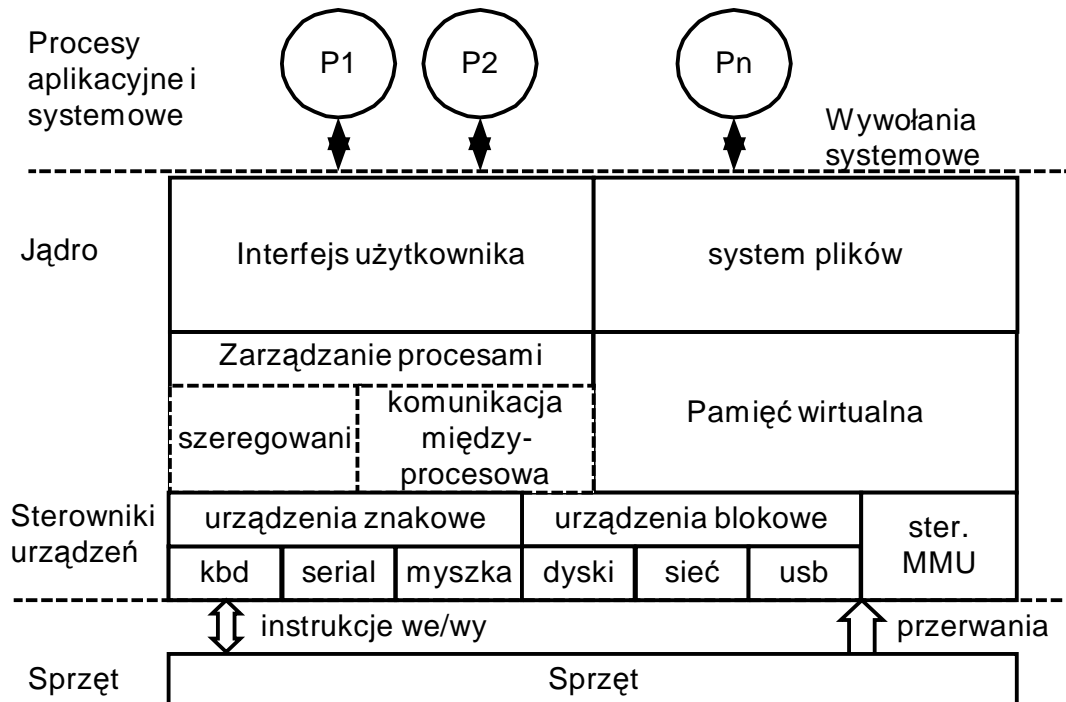
Rys. 2-1 System operacyjny jako interfejs pomiędzy sprzętem a oprogramowaniem

## 2.2 Podstawowe struktury systemów operacyjnych:

- system monolityczny
- system z mikrojądrem.

### 2.2.1 Systemy z jądrem monolitycznym

W systemie monolitycznym podstawowe funkcje systemu umieszczone są w pojedynczym module programowym zwanym jądrem.



Rysunek 2-1 Jądro monolityczne

Fragmenty kodu jądra wykonywane są:

- pod wpływem przerw
- uruchamiania wywołań systemowych.

Cechy systemu z jądrem monolitycznym:

- Jądro nie podlega szeregowaniu
- Awaria w jego obrębie skutkuje awarią całego systemu.
- Wysoka szybkość działania

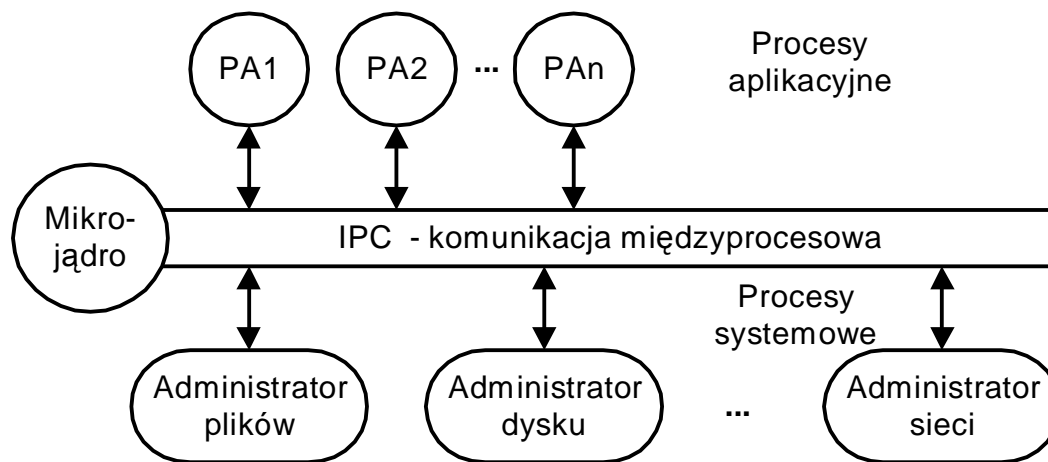
Systemy monolityczne: Linux



## 2.2.2 Systemy z mikrojądrem

Budowa systemu z mikrojądrem:

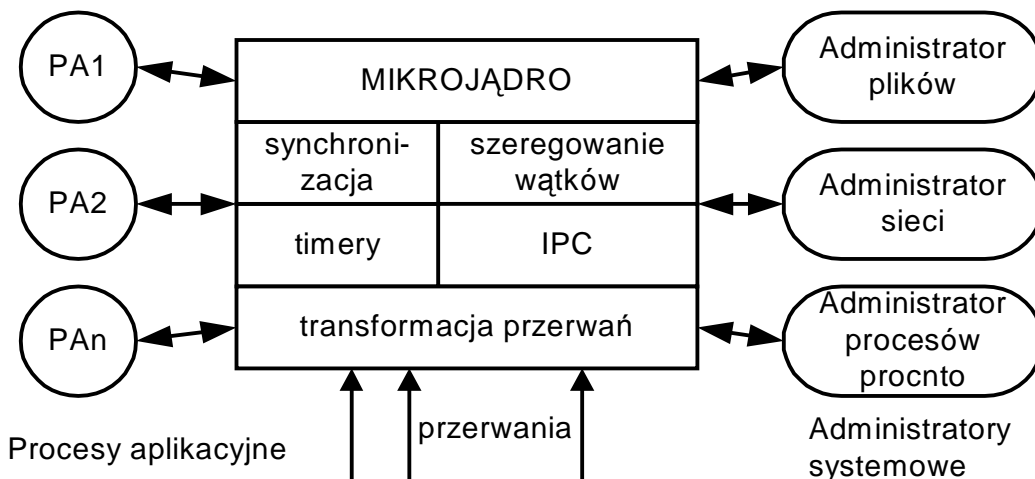
- Mikrojądro (ang. *Microkernel*)
- Procesy systemowe (ang. *System Processes*)
- Procesy aplikacyjne (ang. *Application Processes*)



Rysunek 2-2 Struktura systemu z mikrojądrem (na przykładzie QNX6 Neutrino)

Funkcje mikrojądra:

1. Implementacja podstawowych mechanizmów komunikacji międzyprocesowej: komunikatów, impulsów, zdarzeń, sygnałów.
2. Implementacja funkcji synchronizacji wątków takich jak muteksy, semafony, zmienne warunkowe, bariery, blokady, operacje atomowe.
3. Szeregowanie – procesy i wątki szeregowane są przez mikrojądro zgodnie z dostępnymi algorytmami szeregowania: FIFO, karuzelowy, sporadyczny.
4. Implementacja czasomierzy (ang. *Timers*).
5. Obsługa przerwań.



Rysunek 2-3 Struktura mikrojądra

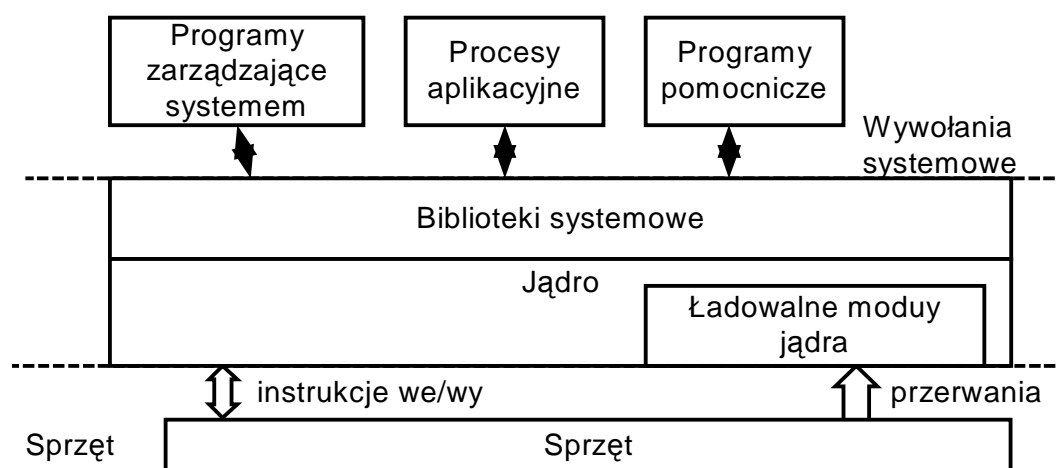
Zalety systemu z mikrojądrem:

1. Niezależne szeregowanie procesów systemowych - obsługa urządzeń wejścia / wyjścia - procesy, które podlegają zwykłemu szeregowaniu. Mechanizm umożliwia osiągnięcie lepszych charakterystyk czasowych systemu - ważne w systemach RTS.
2. Modularność - prowadzi do zwiększenia niezawodności.
3. Wzajemna izolacja procesów - każdy z procesów systemowych wykonywany jest w oddzielnie chronionym segmencie przestrzeni adresowej. Awaria jednego z procesów nie powoduje awarii innego procesu.
4. Możliwość dynamicznego uruchamiania procesów systemowych - procesy systemowe są zwykłymi procesami.

### 3. System Linux

System składa się z trzech głównych elementów

- Jądro – odpowiada za realizację wszystkich istotnych abstrakcji systemu: pamięci wirtualnej, procesów i plików.
- Biblioteki systemowe – określają standardowy zestaw wywołań systemowych za pomocą których aplikacje mogą współdziałać z jądrem.
- Programy i demony systemowe – programy wykonujące funkcje systemowe i pomocnicze. W tym demony systemowe – programy pracujące w sposób ciągły i realizujące różne funkcje (ftp, demon sieciowy, itd.).



Rys. 3-1 Ogólny schemat systemu Linux

Jądro dostarcza wszystkich podstawowych funkcji niezbędnych do wykonywania procesów. Jednak interfejs do usług jądra realizowany jest poprzez biblioteki systemowe. Dostarczają one bardziej złożonych wersji podstawowych usług systemowych.

System Linux zawiera wiele programów działających w trybie użytkownika. Są to narzędzia niezbędne do rozpoczęcia pracy systemu, konfigurowania urządzeń, ładowania modułów jądra a także serwery systemowe. Serwery systemowe obsługują informacje napływające z sieci, inicjują pracę użytkowników itd. W skład systemu wchodzi też programy służące do obsługi systemu plików, graficznego interfejsu użytkownika, kompilatory i inne narzędzia niezbędne do codziennej pracy użytkownika.

## 4. Sieciowe systemy operacyjne

Dziedziny zastosowań:

- Serwery plików
- Serwery bazy danych
- Serwery WWW
- Serwery Aplikacji

Funkcje:

1. Zapewnienie komunikacji międzyprocesowej
2. Implementacja usług nazewniczych
3. Implementacja sieciowego systemu plików
4. Dostarczenie standardowych usług i aplikacji sieciowych: FTP, poczta elektroniczna, WWW, Corba.