# Systemy operacyjne
# 29.10.2010

## System sieciowy  UNIX-a

System sieciowy UNIX-a używa potoku umożliwiającego przepływ strumienia bajtów między dwoma procesami i przepływ gniazdek (sockets) dla procesów powiązanych ze sobą funkcją systemową fork – do pracy w sieci.

- Gniazdko jest punktem końcowym komunikacji o związanym z nim adresie, którego rodzaj zależy od domeny komunikacyjnej gniazdka
- Format adresu domeny UNIX-a jet zwyczajną nazwą ścieżki w systemie plików
- Procesy komunikujące się w domenie Internetu stosują protokoły TCP/IP oraz adresy sieci Internet – 32-bitowy numer komputera macierzystego i 32-bitowy numer portu
- Istnieje kilka gniazdek reprezentujących klasy usług:
    - Strumieniowe – umożliwiają dwukierunkowe i niezawodne przekazywanie sekwencyjnych strumieni danych, przy przesyłaniu żadne dane nie giną ani nie są podwajane nie ma również ograniczeń na rekordy. Gniazdko to występuje w domenie Internet w której używa się protokołu TCP. Gniazdko to występuje także w domenie UNIX-a, potoki implementuje się jako pracy komunikacyjnych gniazdek strumieniowych
    - Pakietów sekwencyjnych – te gniazdka tworzą strumienie danych tak jak w gniazdkach strumieniowych lecz stosuje się tu ograniczenia rekordów – rozdaj gniazdek używanych w sieci Xerox
    - Datagramów – takie gniazdka przesyłają w każdym kierunku komunikaty mające zmienną długość, nie gwarantują że komunikaty dotrą w takim samym porządku, w którym zostały wysłane, że nie będą powtórzone, ani też że nadejdą w ogóle – ten rodzaj gniazdek jest zaimplementowany w domenie sieci Internet przez protokół UDP
    - Surowe – te gniazdka umożliwiają procesom bezpośredni dostęp do protokołów implementujących inne rodzaje gniazdek – na przykład w domenie Internetu jest możliwe osiągnięcie protokołu TCP oraz IP, jak również protokołu sieci Ethernet, ta właściwość jest przydatna przy opracowywaniu nowych protokołów.

---

**Adresowanie pośrednie:**

Tryb pośredni (ang.*indirect mode*) jest to taki tryb adresowania, w którym zawartość komórki określonej przez podstawową informację adresową jest traktowana także jako adres (określenie zgodne z normą PN-71/T-01016). Adresowanie pośrednie może być wielopoziomowe (ang. *multilevel addressing*), tzn. adres $A_1$znajdujący się w komórce $A_0$, wyznaczonej przez informację adresową, nie musi być adresem miejsca przechowywania danych, lecz może być adresem miejsca przechowywania następnego adresu $A_2$, itd. Liczba rozumianych w ten sposób poziomów adresowania jest, oczywiście, ograniczona.

**Połączeniowe i bezpołączeniowe usługi sieciowe**:

Protokoły i obsługiwany przez nie ruch danych w sieci mogą mieć charakter połączeniowy lub bezpołączeniowy. W połączeniowej obsłudze danych używana jest trasa (ścieżka) między stacją nadawczą a stacją odbiorczą, utworzona tylko na czas trwania transmisji. W obsłudze bezpołączeniowej dane przepływają przez połączenie ciągle istniejące.

W usłudze połączeniowej można wyróżnić trzy fazy:

- budowa połączenia,
- przesyłanie danych,
- likwidacja połączenia.

W wyniku budowy połączenia zostaje utworzona pojedyncza ścieżka między stacją nadawczą a stacją odbiorczą. Zasoby sieciowe powinny mieć rezerwę pozwalającą na zapewnienie realizacji usługi, np. zagwarantowaną szybkość przesyłania danych.
W fazie przesyłania dane są transmitowane przez utworzoną ścieżkę w sposób sekwencyjny. Dane docierają do stacji odbiorczej w kolejności ich wysyłania przez stację nadawczą.
W fazie likwidacji połączenia utworzone połączenie ulega przerwaniu. Dalsza transmisja między stacjami nadawczą i odbiorczą musi być ponownie poprzedzona fazą budowy połączenia.
Usługę połączeniową charakteryzują dwie istotne wady w porównaniu z usługą bezpołączeniową:

- statyczny wybór ścieżki,
- statyczna rezerwacja zasobów sieciowych.

Statyczny wybór ścieżki jest kłopotliwy, ponieważ oznacza potrzebę przesyłania całego ruchu przez tę samą statyczną trasę; uszkodzenie jej w dowolnym miejscu powoduje zerwanie całego połączenia. Statyczna rezerwacja zasobów sieciowych stwarza kłopoty, ponieważ wymaga gwarantowanej szybkości transmisji oraz zaangażowania zasobów sieciowych, z których nie mogą korzystać inni użytkownicy sieci.
Należy jednak podkreślić, że połączeniowa usługa jest bardzo przydatna w aplikacjach nie tolerujących opóźnień. Na przykład zastosowania wymagające przesyłania głosu i obrazu są oparte na usługach połączeniowych.
W usłudze bezpołączeniowej nie buduje się jedynej ścieżki między stacją nadawczą i odbiorczą, pakiety do stacji odbiorczej mogą docierać w innej kolejności niż są wysyłane przez stację nadawczą, ze względu na to, że mogą być przesyłane różnymi trasami. W usłudze bezpołączeniowej dane przepływają przez trwałe połączenia między węzłami sieci, a każdy pakiet jest obsługiwany indywidualnie i niezależnie od innych pakietów danego komunikatu. Jest to możliwe pod warunkiem, że każdy pakiet jest kompletnie zaadresowany, to znaczy, że każdy z nich ma swój adres stacji nadawczej i stacji odbiorczej.
Usługa bezpołączeniowa ma dwie zalety:

- dynamiczną selekcję ścieżki (trasy),
- dynamiczny przydział pasma.

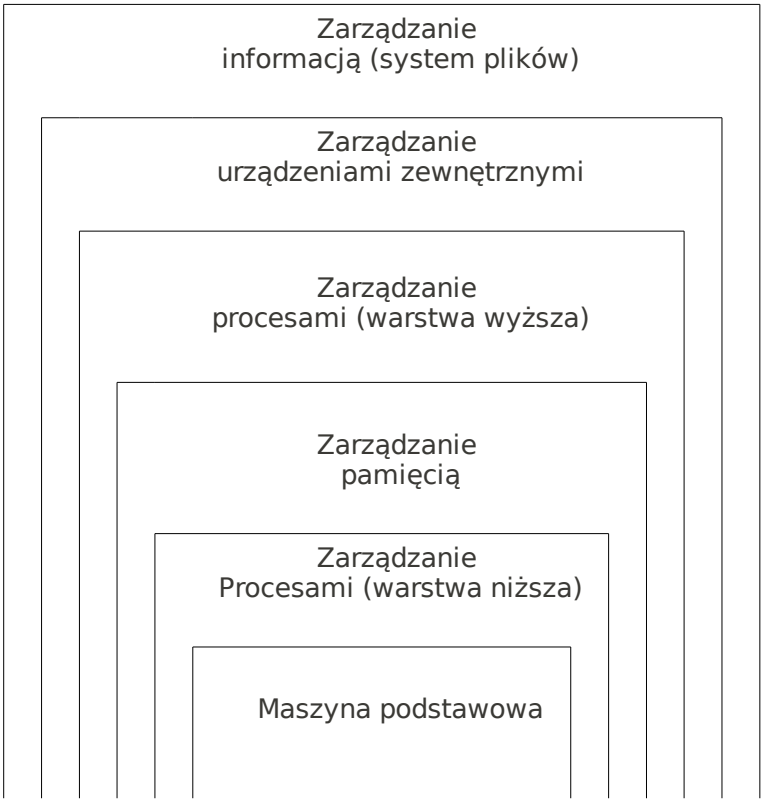
Dynamiczna selekcja ścieżki umożliwia trasowanie z pominięciem uszkodzonego miejsca w sieci dzięki sterowaniu przepływem odnoszącym się do każdego pakietu z osobna. Dynamiczny przydział pasma jest bardzo efektywny, ponieważ pasmu nie przydziela się zasobów, jeśli nie są one używane.
Usługi bezpołączeniowe są zalecane przy transmisji danych w aplikacjach tolerujących pewne opóźnienia i powtórzenia.

**Model odniesienia i warstwy w sieci UNIX**

| Lp. | | Warstwy systemu UNIX | | Przykłady warstw | |
|---|---|---|---|---|---|
| **1** | Zastosowań | Programy użytkowników | | | |
| **2** | Prezentacji | Biblioteki programów | | FTP | TELNET |
| **3** | Sesji | Gniazdka | | Gniazdko strumieniowe | |
| **4** | Transportowa | Protokół | | TCP | |
| **5** | Sieciowa | Protokół | Routing | IP | |
| **6** | Łącza danych | Interfejs sieciowy | | Ethernet | |
| **7** | Sprzętowa | Sprzęt sieciowy | Medium transmisji | Kontroler sieciowy | |

# System operacyjny

## warstwy systemu operacyjnego




Struktura warstwowa jest najczęściej stosowaną w projektowaniu i implementacji jądra systemu operacyjnego. Idea podejścia warstwowego polega na tym, że operacje warstwy wyższej implementowane są z wykorzystaniem usług warstwy niższej.

Maszynę podstawową stanowi sprzęt (procesor, pamięć, interfejs urządzeń zewnętrznych).

Warstwa niższa modułu zarządzania procesami odpowiedzialna jest za planowanie przydziału procesora (szeregowanie zadań) oraz synchronizacje procesów)

Warstwa wyższa modułu zarządzania procesami odpowiedzialna jest za tworzenie i usuwanie procesów oraz za mechanizmy komunikacji pomiędzy procesami.

Zarządzanie pamięcią sprowadza się do przydziału i zwalniania pamięci.

Zarządzanie urządzeniami wejścia/wyjścia obejmuje przydział i zwalnianie tych urządzeń oraz utrzymywanie ich stanu.

Warstwa zarządzania informacją dostarcza podstawowych operacji na plikach (tworzenie, usuwanie, odczyt itp.)

## Elementy składowe jądra systemu operacyjnego

- Moduł zarządzania procesami
- Moduł zarządzania pamięcią operacyjną
- Moduł zarządzania plikami
- Podsystem wejścia/wyjścia
- Moduł zarządzania pamięcią podręczną
- Interfejs sieciowych
- Podsystem ochrony*
- Interpreter poleceń (?)

(?) - Interpreterów może być wiele, mogą być zmieniane, mogą mieć inne podejścia technologiczne, tekstowe, graficzne, foniczne.

* - Ochrona informacji jest realizowana na wielu szczeblach prac programistycznych.

### Moduł zarządzania procesami

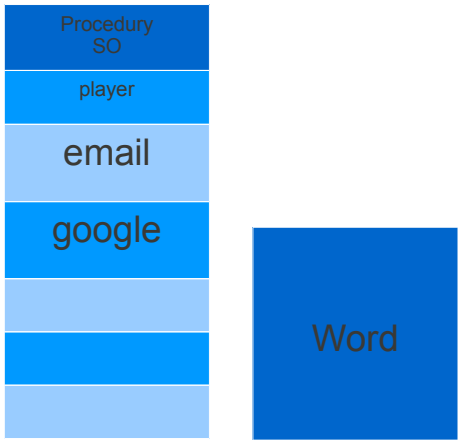
- tworzenie i usuwanie procesów
- wstrzymywanie i wznawianie procesów (przełączanie kontekstu)
- planowanie przydziału procesora (szeregowanie procesów)
- dostarczanie mechanizmów synchronizacji i komunikacji procesów
- dostarczanie mechanizmów obsługi zakleszczeń

Deadlock – załóżmy, że w systemie operacyjnym są 2 procesy: proces A i proces B.
A potrzebuje ingerencji w dane dysku, następnie chce wykonać obliczenia, a potem wydrukować wyniki. B najpierw chce coś wydrukować, potem wykonać obliczenia, a następnie ingerować w dane dysku. Systemy operacyjny ma jedną jednostkę dyskową i jedną drukarkę. Procesy A i B zaczynają się i otrzymują kwant czasu. Proces A otrzymuje dysk, a proces B drukarkę, obydwa procesy trzymają to co otrzymały aż skończą swoje działanie. Nie mogą otrzymać tego czego potrzebują w dalszej części swojego działania (wzajemnie się blokują) i system się zawiesza.

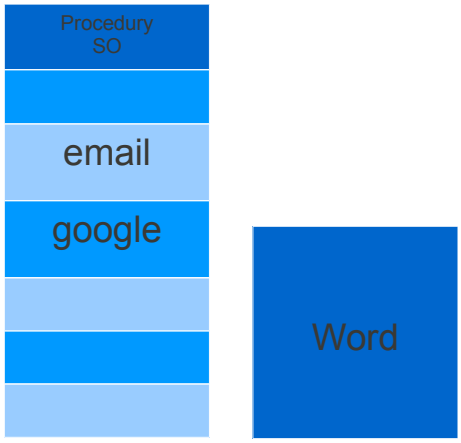
Starvation – załóżmy w systemie operacyjnym mamy 5 jednostek dyskowych. Przychodzi ciąg zadań, jedno z nich chce 3 jednostek dyskowych, lecz dostaje tylko 2 bo 3 są już używane i cały czas czeka na trzeci dysk, jednak w trakcie czekania nadchodzi kolejny strumień ogromnej ilości małych zadań, z których każdy potrzebuje po jednym dysku, i przejmują każdy zwolniony dysk. 2 dyski zajęte przez oczekujący proces są ciągle w stanie zawieszenia.

**Moduł zarządzania pamięcią operacyjną**

- przydzielanie i zwalnianie obszarów pamięci
- ochrona pamięci – utrzymywanie informacji o stanie zajętych obszarów pamięci (prawa dostępu, właściciel itp)
- realizacja wymiany procesów
- realizacja wymiany stron w przypadku wirtualizacji




Jesteśmy użytkownikami systemu operacyjnego, uruchomiliśmy programy: player, email, google, ale chcielibyśmy uruchomić, także Word, niestety mamy za mało pamięci operacyjnej by to zrobić... Wyłączamy więc playera...




Niestety nie rozwiązało to naszego problemu, nie możemy podzielić programu word na dwie części mimo że by się w nich zmieścił... gdybyśmy jednak przesunęli email i google w pamięci (re-lokowali) tak aby wolne przestrzenie pamięci scaliły się.. to udało by się nam załadować worda... I tu pojawia się problem **relokacji.** Jak przenieść program w inną przestrzeń pamięci, przecież program będzie odwoływał się do starych adresów zmiennych i procedur a tam już będzie coś innego. Są tutaj dwa podejścia.

**Relokacja statyczna** – polega na przeadresowaniu każdego adresu występującego w programie (każdej zmiennej, każdej procedury) – stary sposób

**Relokacja dynamiczna** - program napisany jest tak by być całością, zaczyna się on adresem 0 bazą (base), zmienne i procedury są adresowane w stosunku do początkowego adresu – zera, ta odległość między bazą a jakimś bytem to offset. (w skrócie adres zmiennej, procedury itd. to po prostu base+offset ). Relokacja dynamiczna polega więc na przeniesieniu całego programu przez zmianę bazy (base). Dla przykładu program znajduje się na początku pamięci operacyjnej. Zaczyna się w komórce o adresie : 0, zmienna int A ma adres 14 , gdy przesuwamy program tak aby zaczynał się w komórce 1000 to A dalej ma adres 14 (dla programu), jednak w momencie odwołania się do A w ostatniej chwili do adresu zostaje dodany 1000.