Akademia Nauk Stosowanych
Wydział Nauk Inżynieryjnych
Kierunek: Informatyka
studia I stopnia, semestr 2

AKADEMIA NAUK STOSOWANYCH W NOWYM SĄCZU

Systemy operacyjne

WYKŁAD 1

dr inż. Stanisława Plichta

splichta@ans-ns.edu.pl

LINUX – zarządzanie pamięcią

- W systemie Linux zarządzanie pamięcią obejmuje:

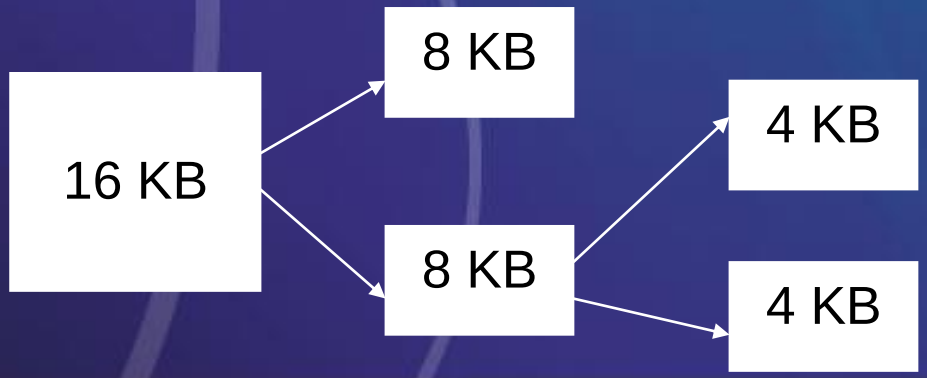
- **System zarządzania pamięcią fizyczną**
- **System zarządzania pamięcią wirtualną**

LINUX – zarządzanie pamięcią

- Zarządcą podstawowej pamięci fizycznej w jądrze systemu jest dyspozytor stron - przydział i zwalnianie wszystkich fizycznych stron (ramek).

ALGORYTMEM SĄSIEDNICH STERT

- Najmniejszą możliwą do przydzielenia w ten sposób jednostką jest pojedyncza strona fizyczna.



LINUX – zarządzanie pamięcią

- Wszystkie przydziały pamięci są zarezerwowane
  - statycznie
  - dynamicznie

Specjalizowane podsystemy zarządzania pamięcią

- System pamięci wirtualnej.
- Dyspozytor obszarów zmiennej długości, czyli funkcja kmalloc.
- Dwie trwałe pamięci podręczne jądra:
  - podręczna pamięć buforów
  - podręczna pamięć stron

System pamięci wirtualnej

- Odpowiada za opiekę nad przestrzenią adresową widoczną dla każdego procesu.
- Tworzy strony pamięci wirtualnej na żądanie i zarządza sprowadzaniem ich z dysku lub ich wynoszeniem z powrotem na dysk.
- Zarządca pamięci wirtualnej utrzymuje dwa osobne obrazy przestrzeni adresowej procesu:
  - zbiór oddzielnych obszarów
  - zbiór stron

System pamięci wirtualnej

Jądro utworzy nową wirtualną przestrzeń adresową:

- Gdy proces rozpoczyna wykonanie nowego programu za pomocą funkcji systemowej execl - proces otrzymuje nową pustą wirtualną przestrzeń adresową
- Przy tworzeniu nowego procesu za pomocą funkcji systemowej fork - nowy proces dziedziczy przestrzeń adresową procesu macierzystego

System pamięci wirtualnej

- Zmodyfikowana wersja algorytmu zegarowego (drugiej szansy) - wieloprzebiegowy zegar - miara stopnia aktywności strony w ostatnim czasie.
- procedura stronicująca wybiera do wyrzucenia strony wg kryterium najrzadszego ich używania (LRU).
- przydział bloków na urządzeniach wymiany odbywa się wg mapy bitowej używanych bloków - stale przechowywana w PAO.
- Strony zapisywanie w sposób ciągły w sąsiednich blokach - algorytm najlepszego dopasowania.

Rodzaje urządzeń wejścia/wyjścia

1. Urządzenia składowania danych
2. Urządzenia transmisji danych
3. Urządzenia do komunikacji z człowiekiem
4. Urządzenia specjalizowane
   - układy sterowania
   - kasy i drukarki fiskalne itp.
   - urządzenia medyczne

Właściwości urządzeń wejścia/wyjścia

1. **Tryb transmisji danych**
   - znakowy
   - blokowy
2. **Sposób dostępu do danych**
   - sekwencyjny
   - swobodny
3. **Tryb pracy urządzenia**
   - synchroniczny
   - asynchroniczny
4. **Tryby współdzielenia**
   - wyłączny
   - współdzielony
5. **Szybkość działania (transmisji)**
   - od bardzo wolnych (np. drukarka)
   - do bardzo szybkich (np. dysk)
6. **Kierunek dostępu do danych**
   - urządzenia we/wy
   - urządzenia we
   - urządzenia wy

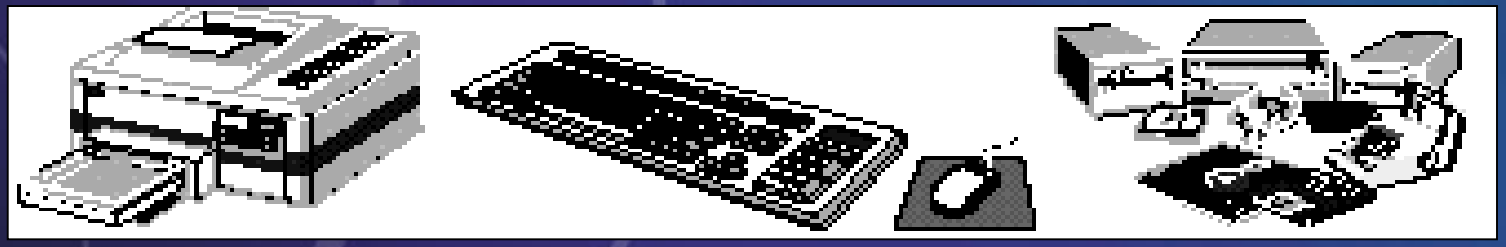
Struktura mechanizmu wejścia/wyjścia

oprogramowanie

sprzęt

Podsystem we/wy w jądrze systemu operacyjnego

Moduł sterujący	Moduł sterujący	Moduł sterujący
sterownik portu		Adapter
Sterownik urządzenia	Sterownik urządzenia	Sterownik urządzenia

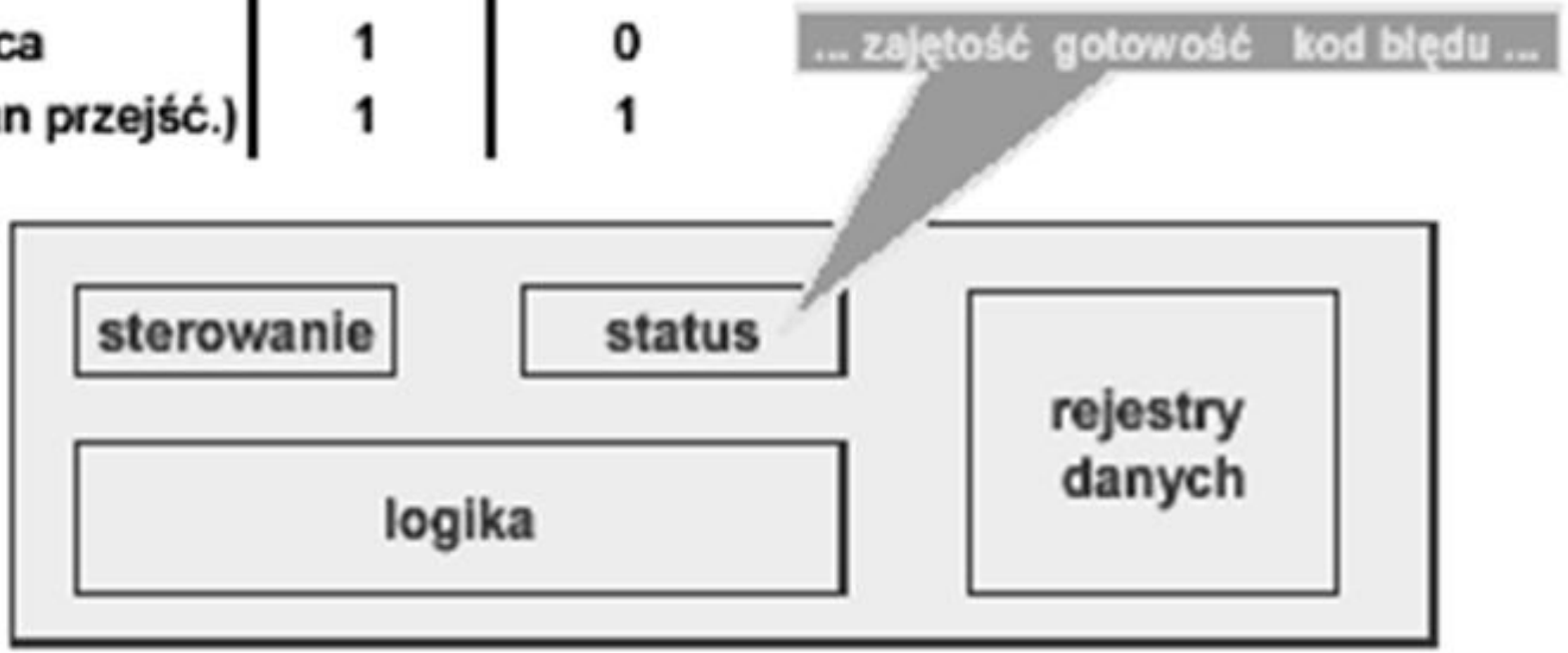


Podsystem wejścia/wyjścia nadzoruje

- Zarządzanie przestrzenią nazw plików i urządzeń.
- Przebieg dostępu do plików i urządzeń.
- Poprawność operacji (np. modem nie może przeszukiwać).
- Przydzielanie miejsca w systemie plików.
- Przydział urządzeń.
- Buforowanie, przechowywanie podręczne oraz spooling.
- Planowanie operacji WE/WY.
- Doglądanie stanu urządzeń, obsługę błędów oraz czynności naprawcze po awarii.
- Konfigurowanie i wprowadzanie w stan początkowy modułu sterującego.

Sterownik portu (adapter)

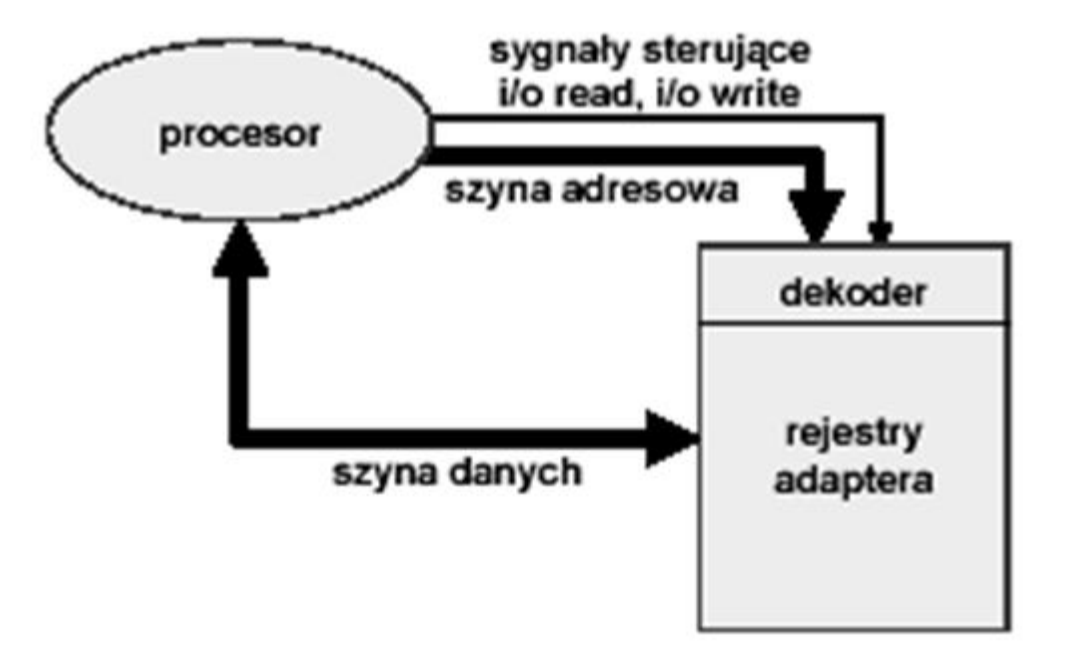
	zajętość	gotowość
bezczynność	0	0
zakończenie	0	1
praca	1	0
(stan przejść.)	1	1



Miejsce urządzeń WE/WY w architekturze systemu komputerowego

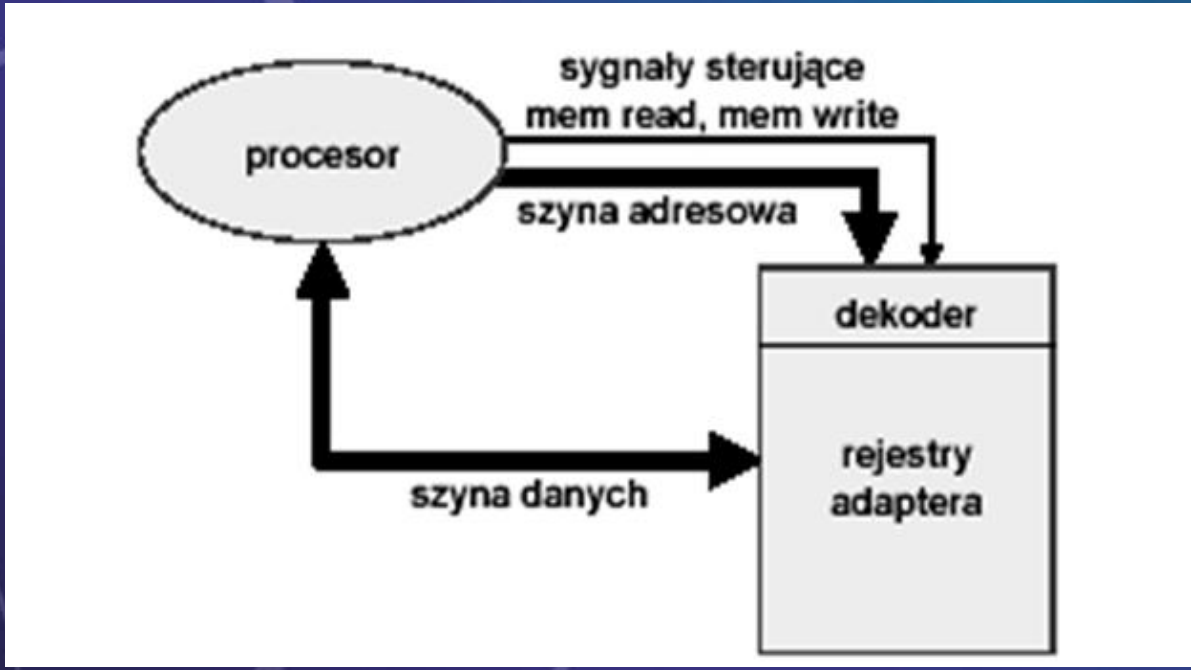
Odwzorowanie w przestrzeni adresowej we/wy

izolowane we/wy



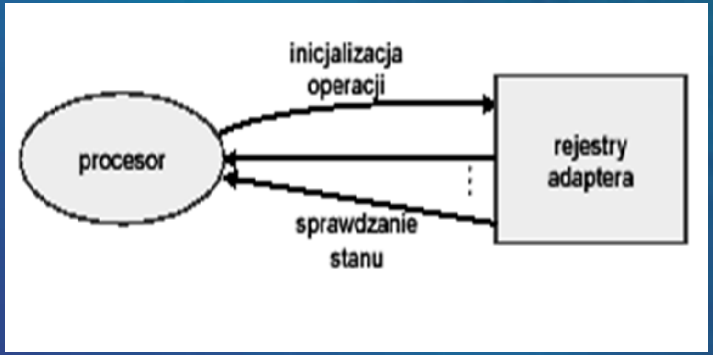
Miejsce urządzeń WE/WY w architekturze systemu komputerowego

Odwzorowanie w przestrzeni adresowej pamięci - rejestry sterownika widoczne są w przestrzeni adresowej pamięci fizycznej



Interakcja jednostki centralnej ze sterownikiem urządzenia we/wy

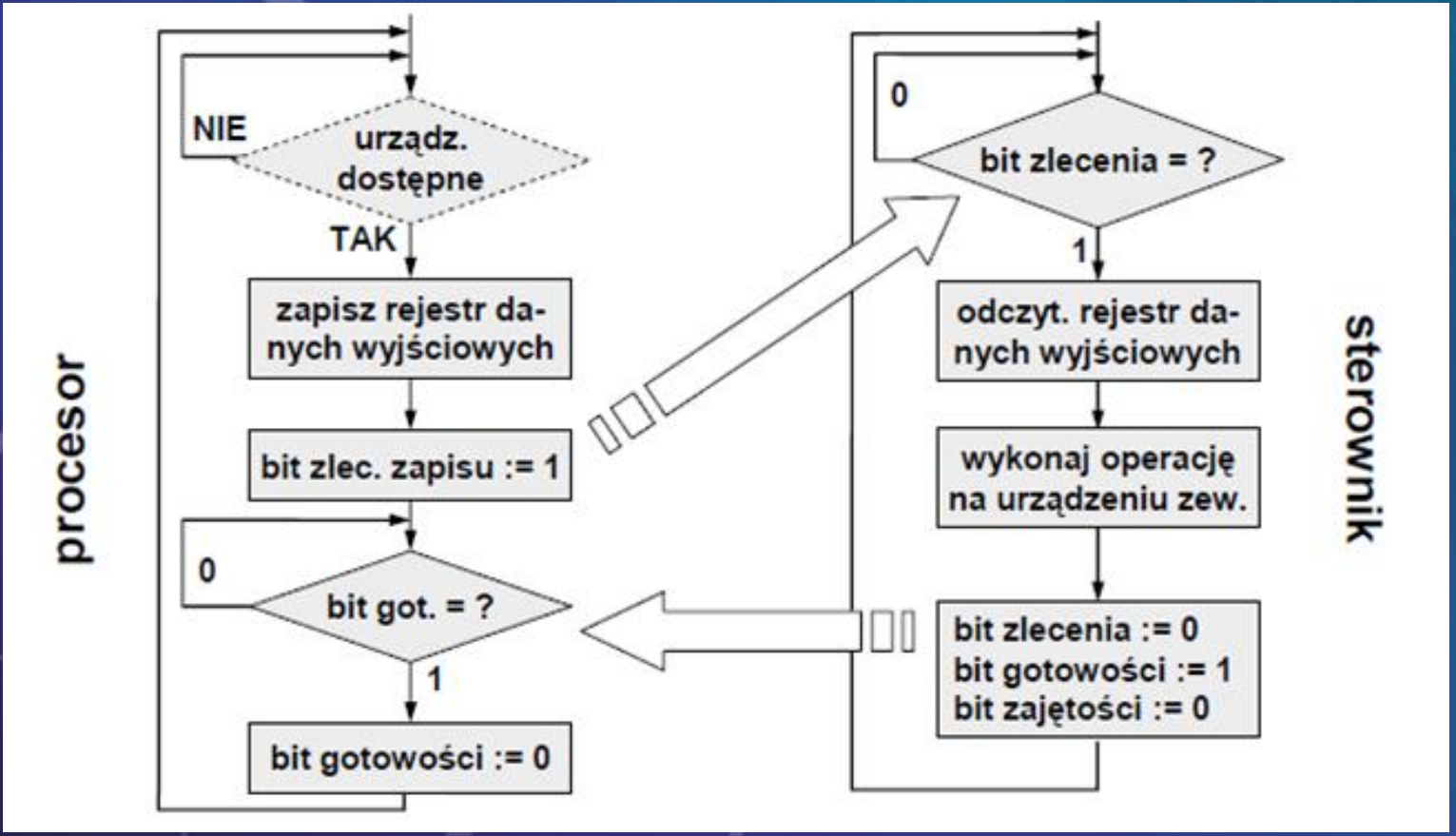
1. Odpytywanie



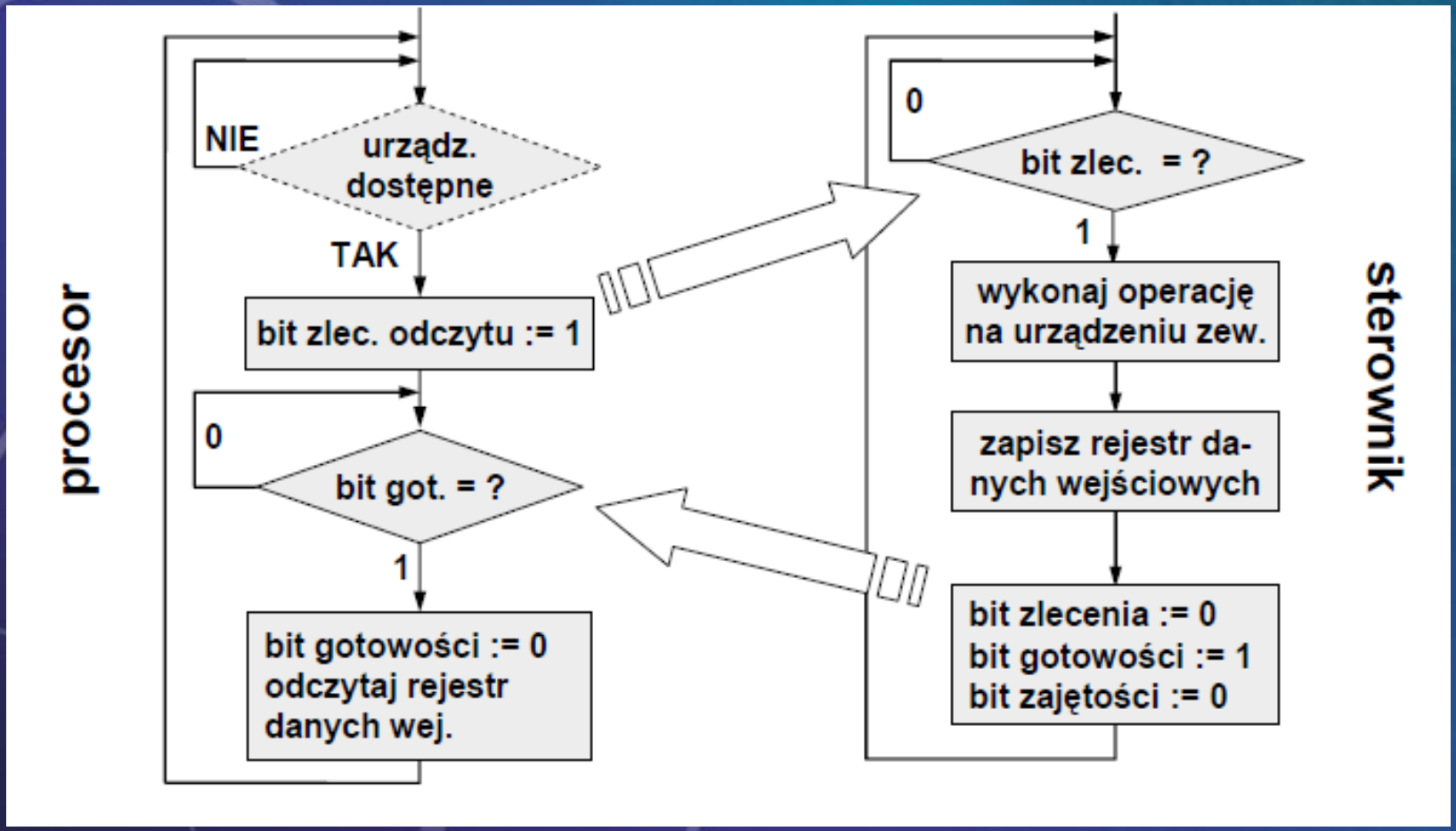
2. Sterowanie przerwaniami

3. Bezpośredni dostęp do pamięci

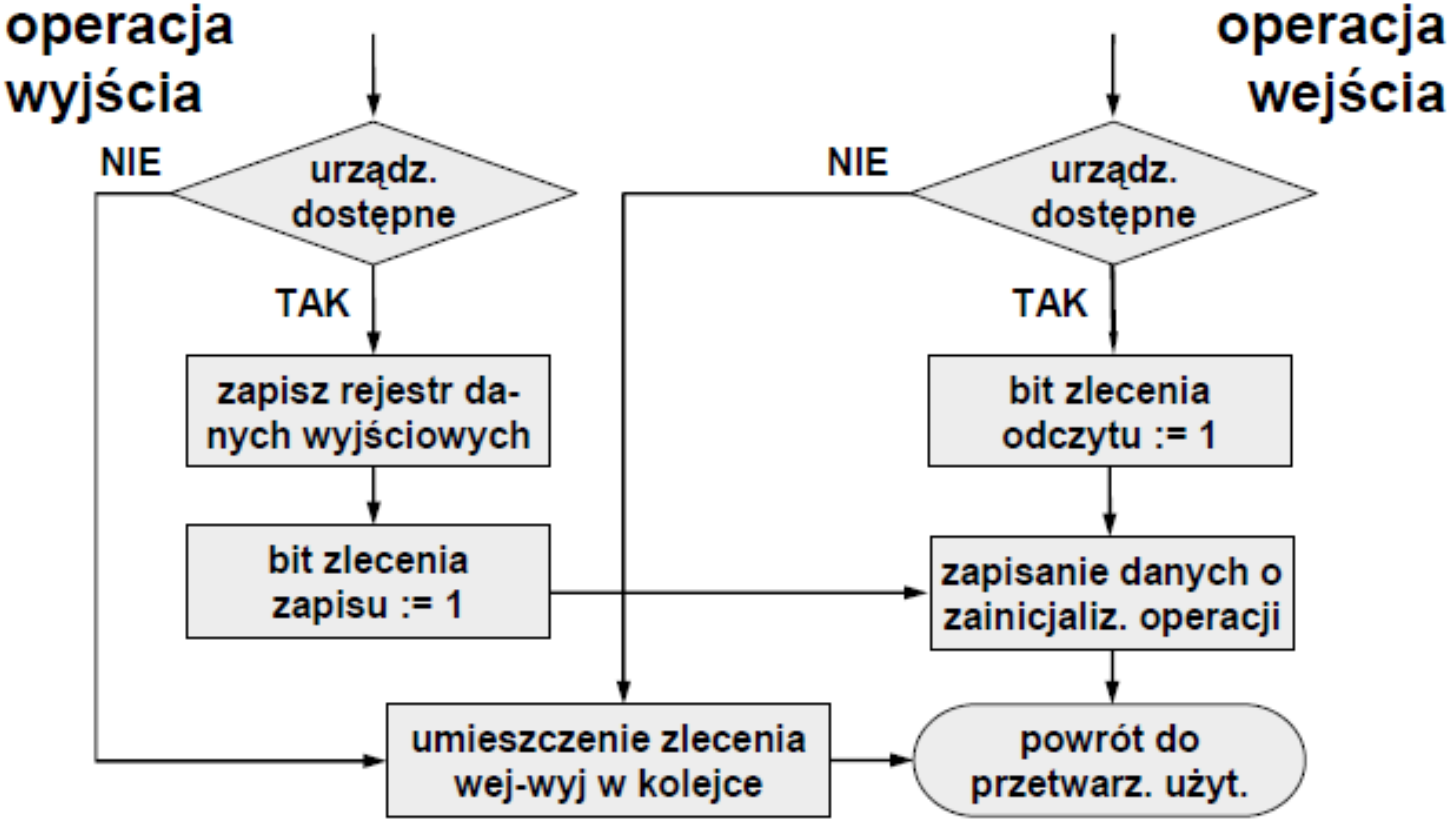
Interakcja procesor – sterownik w operacji wejścia (tryb odpytywania)



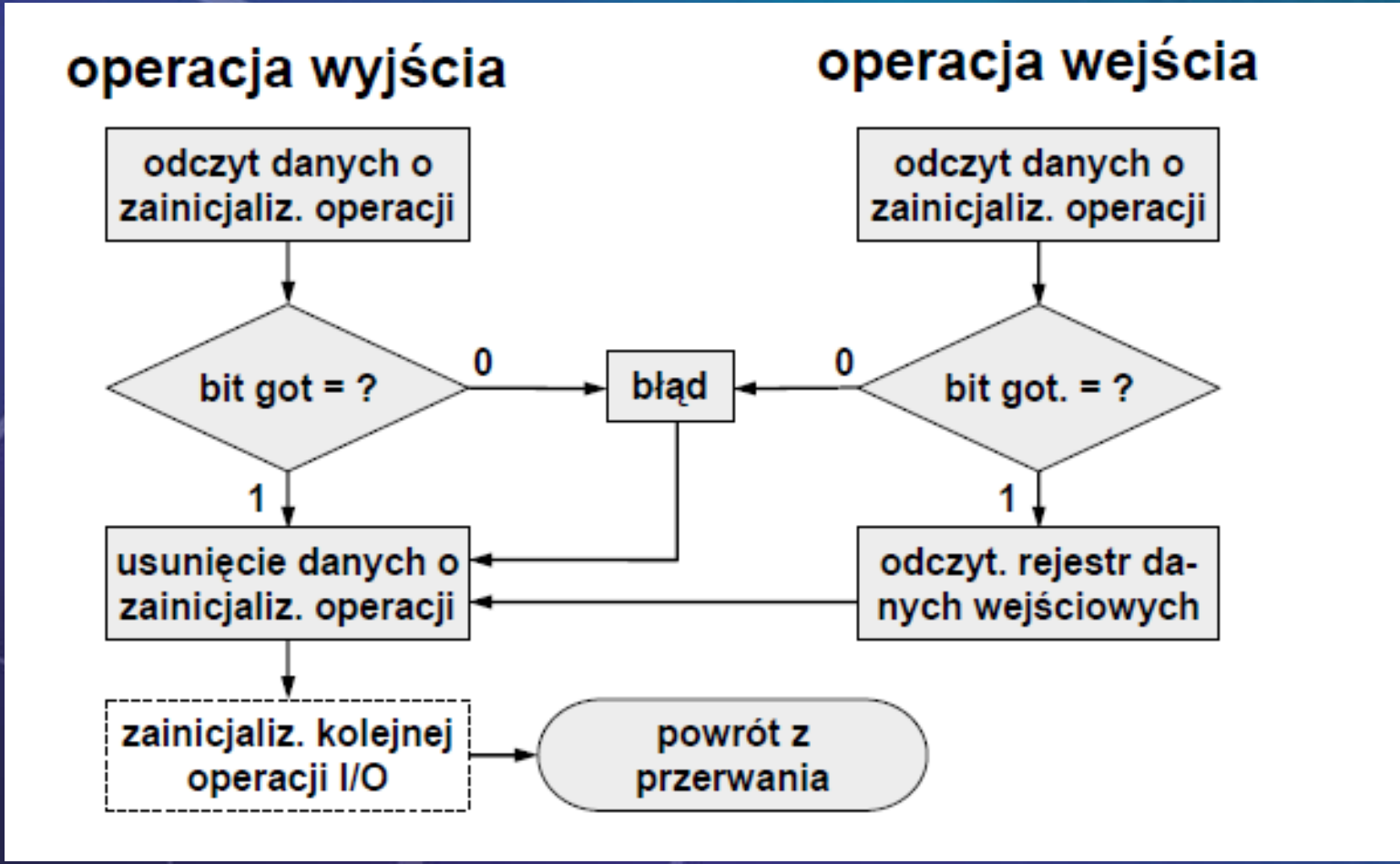
Interakcja procesor – sterownik w operacji wyjścia (tryb odpytywania)



Obsługa sterowana przerwaniami
zlecenie operacji



Obsługa sterowana przerwaniami
reakcja na przerwanie



Obsługa przerwań wielokrotnych

Problem przerwań wielokrotnych polega na zgłoszeniu kolejnego przerwania w czasie obsługi innego przerwania

Podejście do obsługi przerwań wielokrotnych:

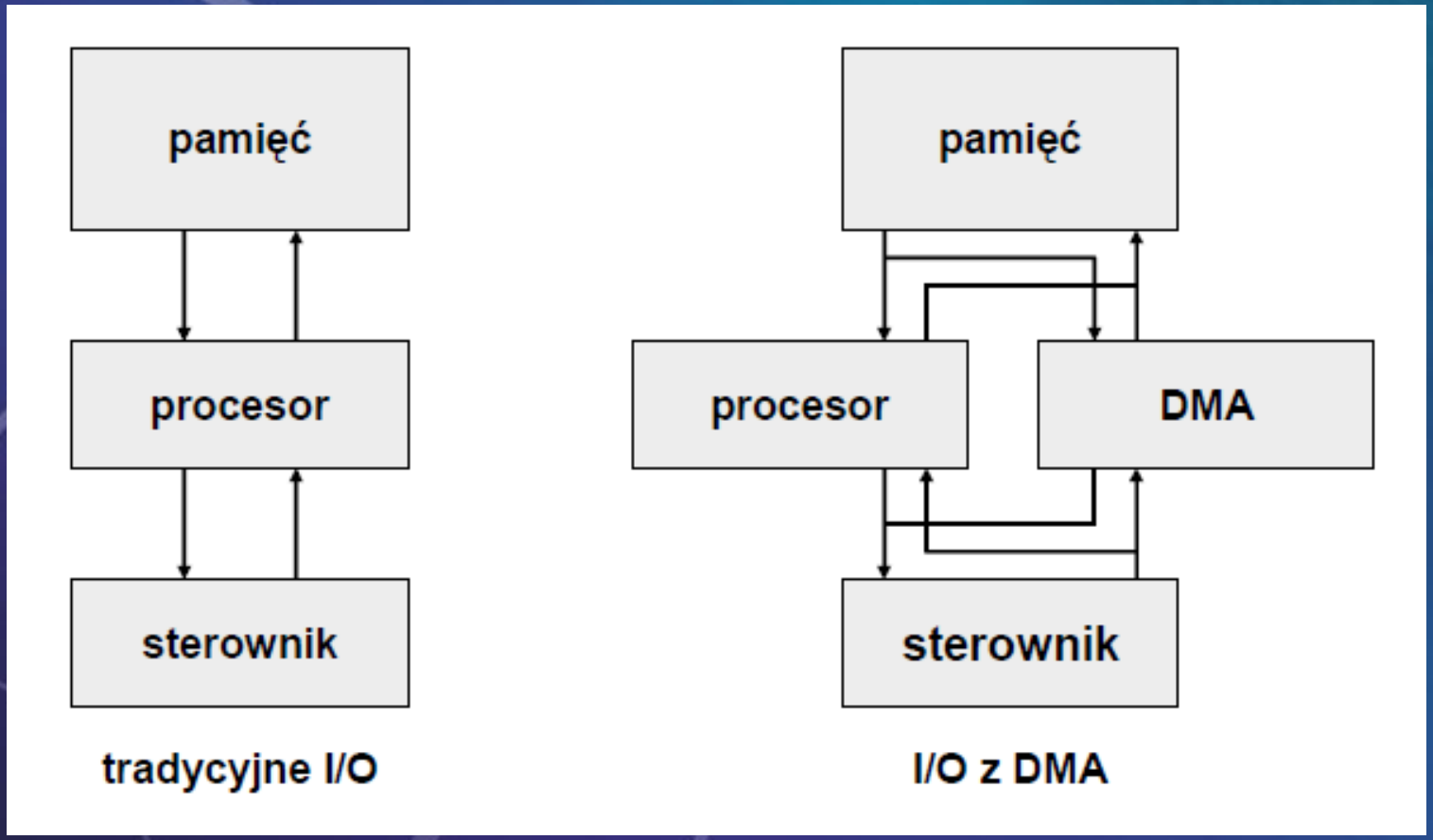
- obsługa sekwencyjna
- obsługa zagnieżdżona
- obsługa priorytetowa

Obsługa sterowana przerwaniami
reakcja na przerwanie

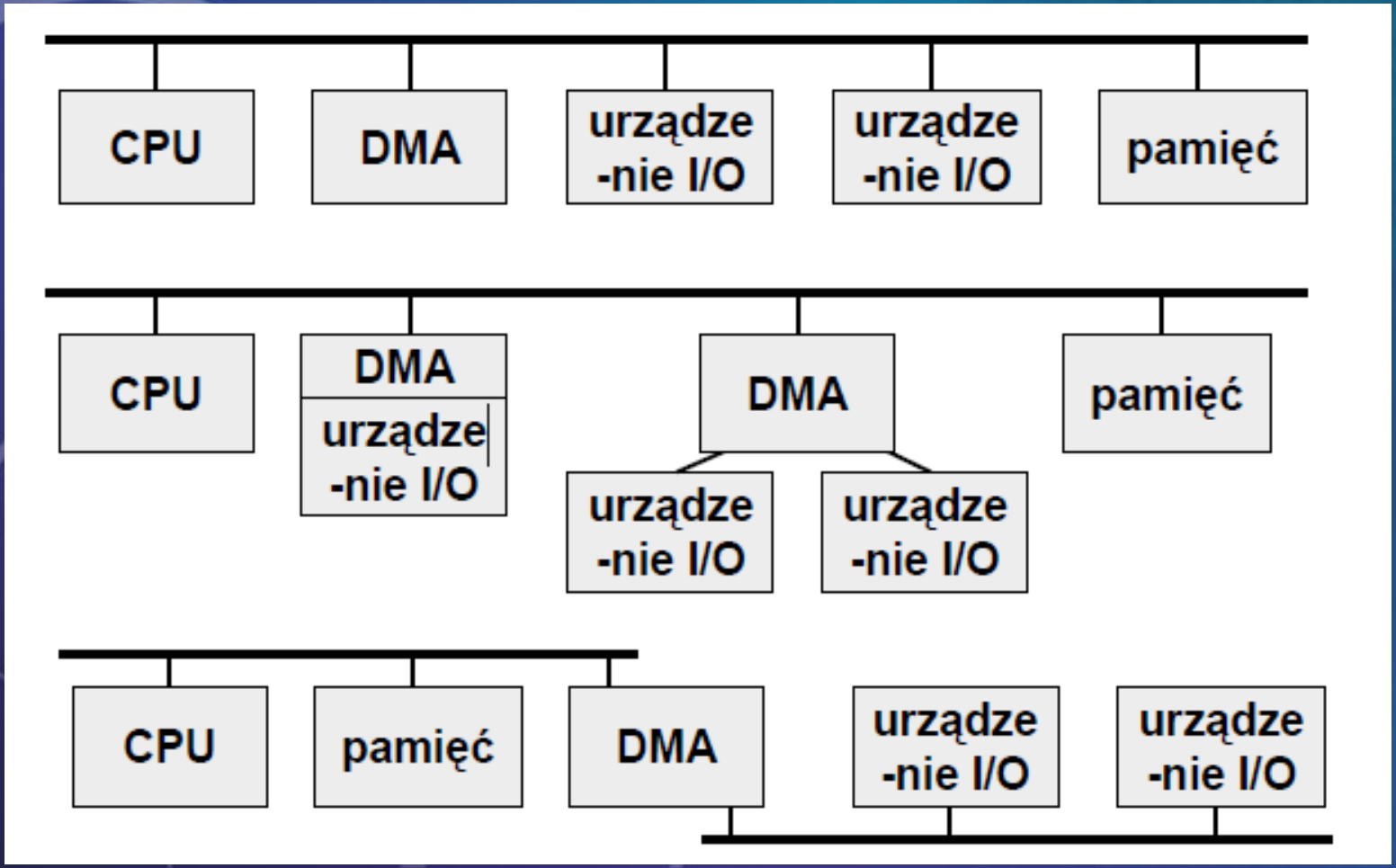
Sposoby identyfikacji źródła przerwania

- Wiele linii przerwań
- Odpytywanie
- Odpytywanie sprzętowe
- Arbitraż na magistrali

Bezpośredni dostęp do pamięci



Organizacja WE/WY



Bezpośredni dostęp do pamięci odbywa się wg następującego scenariusza:

- Moduł sterujący urządzenia dostaje zlecenie przesłania danych pod adres X.
- Moduł sterujący urządzenia zleca sterownikowi urządzenia pobranie danych i przesłanie ich do bufora pod adresem X.
- Sterownik urządzenia rozpoczyna przekaz DMA.
- Sterownik urządzenia przesyła poszczególne bajty do sterownika DMA.
- Sterownik DMA umieszcza otrzymane bajty w pamięci operacyjnej.
- Sterownik DMA wywołuje przerwanie procesora po otrzymaniu wszystkich bajtów.

autor: dr inż. Anna Plichta

Wejście/wyjście

Wejście/wyjście z blokowaniem i bez blokowania

- Blokujące wejście/wyjście
- Nieblokujące wejście/wyjście
- Asynchroniczne wejście/wyjście

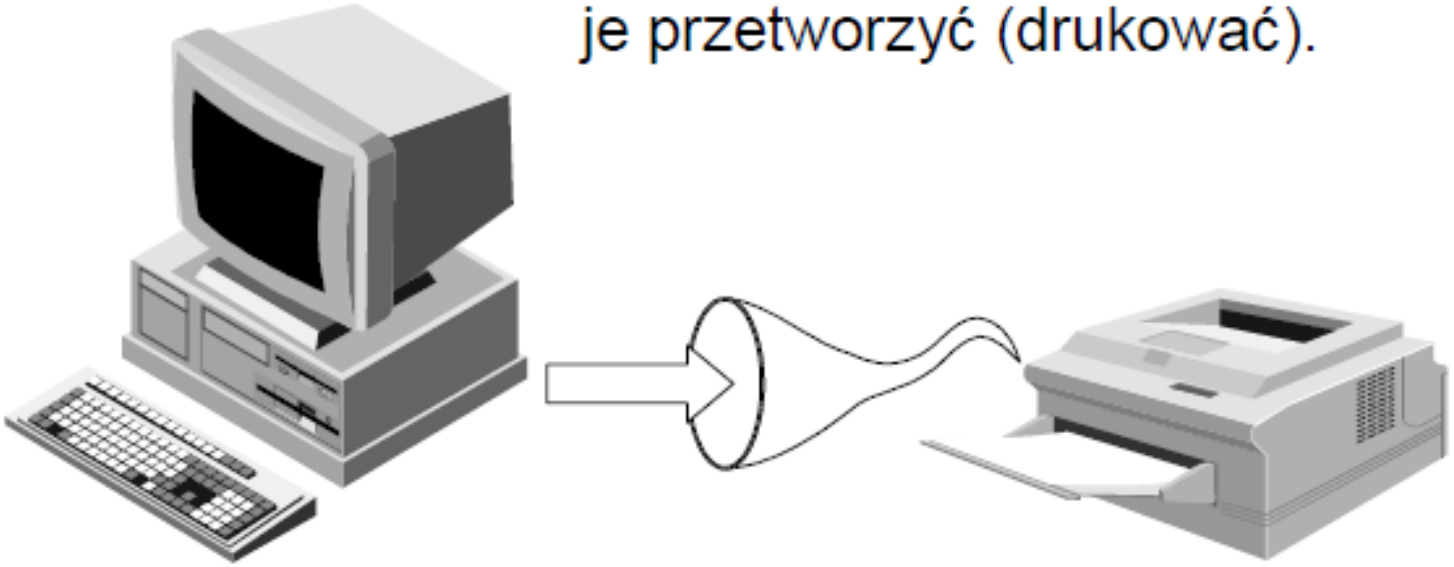
Buforowanie wejścia/wyjścia

- Dopasowanie różnic szybkości
- Dopasowanie jednostek transmisji
- Semantyka kopii

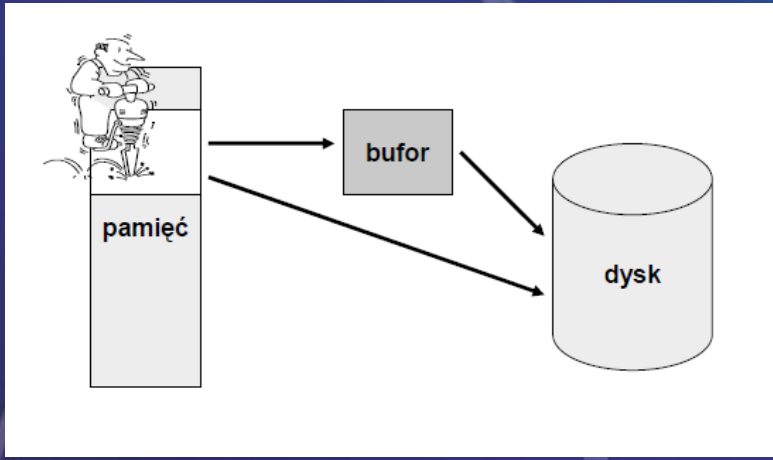
Bufor to obszar pamięci do przechowywania danych przesyłanych między dwoma urządzeniami

Dopasowanie różnic szybkości

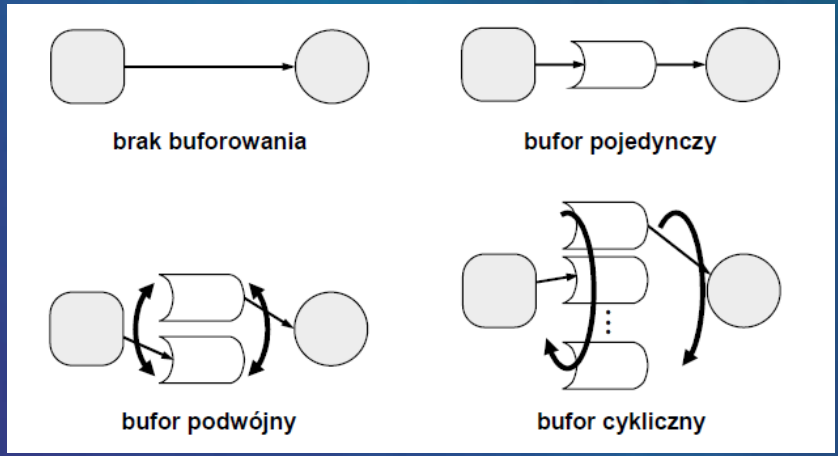
Przykład: komputer potrafi przekazać dane znacznie szybciej niż drukarka je przetworzyć (drukować).



Wejście/Wyjście



Semantyka kopii



realizacja buforowania

Wejście/Wyjście

Przechowywanie podręczne

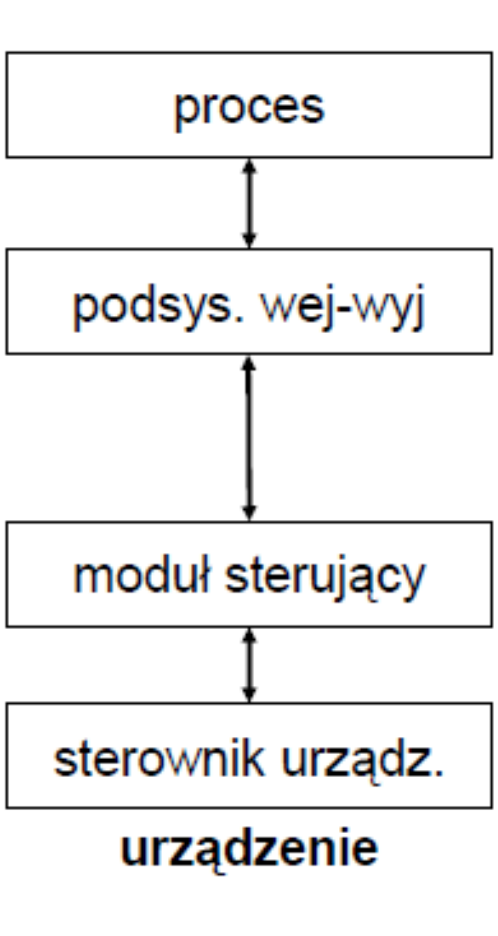
- *Pamięć podręczna* (ang. *cache*) jest obszarem szybkiej pamięci do przechowywania kopii danych
- To udogodnienie pozwala na szybszy dostęp do oryginału danych i jest kluczem do szybszego działania systemu komputerowego

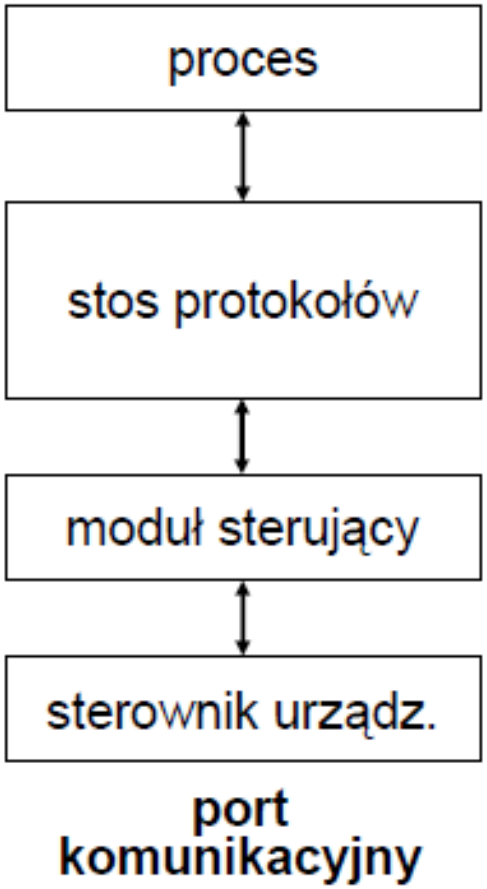
Spooling

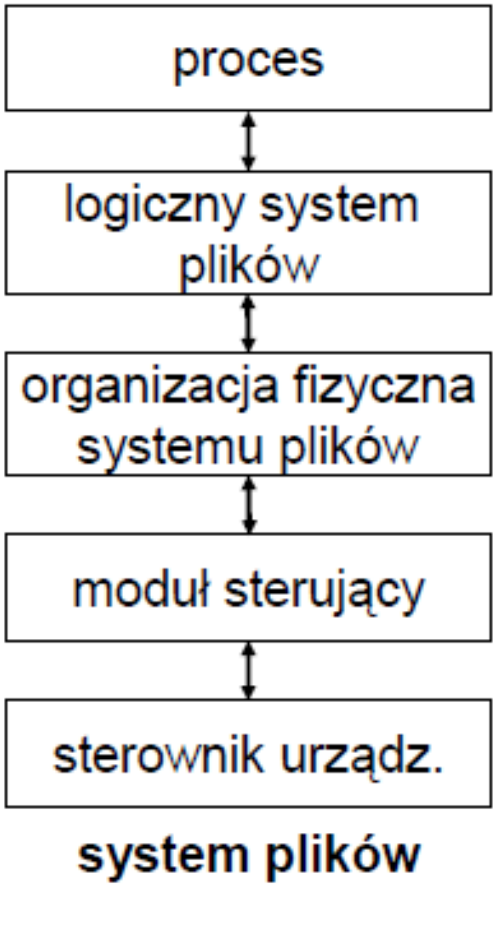
Spooling (SPOOL = *sequential peripheral operation on line* = jednoczesna bezpośrednia praca urządzeń)

Przykładem takiego urządzenia jest drukarka

Wirtualne wejście/wyjście







Obsługa urządzeń wejścia/wyjścia

- System operacyjny musi radzić sobie z rozmaitymi rodzajami błędów:
  - błędny odczyt z dysku,
  - awaria urządzenia (chwilowa lub trwała),
  - chwilowe problemy z zapisem.
- Nieudaną operację można na przykład powtórzyć
- Jeśli nic się nie da zrobić, wywołanie systemowe zwraca kod błędu
- W dzienniku systemowym system operacyjny zapisuje informacje o wszelkich awariach

Obsługa urządzeń wejścia/wyjścia

Rozważmy operację odczytu pliku z dysku

- Ustalenie urządzenia, na którym znajduje się ten plik
- Przetłumaczenia nazwy tego urządzenia na wewnętrzny identyfikator.
- Fizyczny odczyt danych z dysku do bufora
- Udostępnienie danych zlecającemu procesowi
- Przekazanie sterowania temu procesowi

Obsługa urządzeń wejścia/wyjścia

- Proces użytkownika: zlecam odczyt bloku
- Jądro: sprawdzam, czy można już zrealizować zlecenie (np. dzięki **pamięci podręcznej**)
- Jądro: jeśli tak, to udostępniam wynik zlecenia. Jeśli nie, to wysyłam zlecenie do modułu sterującego urządzenia
- Moduł sterujący: wydaję polecenia sterownikowi urządzenia
- Sterownik: steruję urządzeniem, przerywam po zakończeniu operacji wejścia/wyjścia
- Sterownik: generuję **przerwanie**
- Procedura obsługi przerwania: składam dane w buforze.
- Moduł sterujący: ustalam, którą operację wejścia/wyjścia zakończono, informuję jądro o zmianie stanu operacji
- Jądro: przekazuję dane procesowi użytkownika
- Proces użytkownika: mam dane lub wiem, że był błąd

Wydajność

- Wejście/wyjście ma szczególnie duży wpływ na wydajność systemu operacyjnego, ponieważ:

  – wymaga zasobów procesora do wykonywania modułu sterującego i kodu podsystemy wejścia/wyjścia w jądrze,
  – powoduje przełączenia kontekstu w związku z obsługą przerwań (albo aktywne czekanie przy programowanym wejściu/wyjściu),
  – oznacza kopiowanie danych

Wydajność

Wydajność - użytkownik wciska klawisz

- Sterownik klawiatury generuje przerwanie.
- Aktywują się kolejno moduł sterujący klawiatury i podsystem jądra.
- Następuje przełączenie kontekstu na proces użytkownika, który zleca wysłanie wpisane znaku przez sieć do zdalnej maszyny.
- konstruowany jest pakiet sieciowy i przekazywany do modułu obsługi karty sieciowej.
- Wszystko musi przejść znów przez obsługę przerwania i przełączenie kontekstu.
- Pakiet jest odbierany na maszynie zdalnej, gdzie powoduje przerwanie i obsługę.
- Z pakietu wyjmuje się wpisany znak i przekazuje do procesu aplikacyjnego, który ma ten znak zrozumieć i obsłużyć.

Poprawa wydajności

- Poprawę wydajności wejścia/wyjścia osiągniemy poprzez:
- zmniejszenie liczby przełączeń kontekstu,
- zmniejszenie zakresu kopiowanie danych,
- ograniczenie liczby przerwań poprzez wykonywanie operacji wejścia/wyjścia na większych blokach danych, wykorzystanie inteligentnych sterowników i rozsądne wykorzystanie **odpytywanie**,
- wykorzystanie bezpośredni dostęp do pamięci
- zwiększenie równoległości poprzez wykonywanie elementarnych działań za pomocą sprzętu w sterownikach urządzeń,
- zrównoważenie wydajności procesora, pamięci, szyn i operacji wejścia wyjścia.

Realizacja funkcji we/wy

Funkcje urządzeń wejścia/wyjścia można zrealizować w następujących miejscach:

- w kodzie aplikacji (oprogramowanie),
- w kodzie jądra (oprogramowanie),
- w kodzie modułu sterującego (oprogramowanie),
- w kodzie sterownika (sprzęt),
- w kodzie urządzenia (sprzęt).

Bezpieczeństwo i ochrona

Kontrola dostępu programów, procesów i użytkowników do zasobów zdefiniowanych przez system komputerowy

- środki specyfikujące rodzaje wymaganej ochrony (polityka)
- środki ich wymuszania (mechanizmy)

Miarą zaufania do systemu - ochrona i bezpieczeństwo

Ochrona

- System komputerowy - zbiór procesów i obiektów.
- Rodzaj wykonywanych operacji - zależny od obiektu.
- Proces ma dostęp tylko do tych zasobów, do których został uprawniony.

Zasada wiedzy koniecznej (need to know)

Struktura domenowa

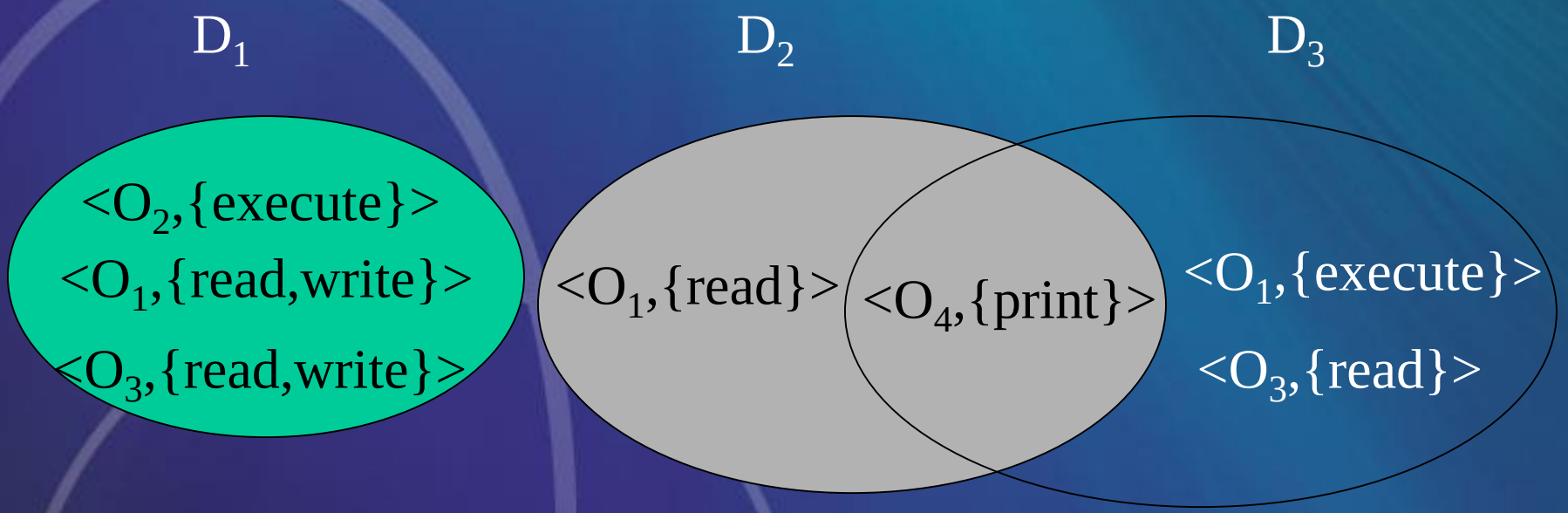
Domena ochrony - definiuje zbiór obiektów i rodzaje operacji, które można na nich wykonywać (prawa dostępu)

<nazwa obiektu, zbór praw>

np. prawo w domenie D:

<plik F, {czytaj, pisz}>

Struktura domenowa



Przykład systemu z trzema domenami
domeny mogą dzielić prawa dostępu

Struktura domenowa

Związek między procesem i domeną:

- statyczny (ustalony)
  - Jeśli chcielibyśmy przestrzegać zasady wiedzy koniecznej to musi istnieć mechanizm zmiany zawartości - (zachowanie minimum niezbędnych praw dostępu).
- dynamiczny
  - Możliwość przełączania procesu między domenami,
  - Możliwość zmiany zawartości domeny,
  - Możliwość utworzenia nowej domeny ze zmienioną zawartością.

Sposoby realizacji domeny

Domeną może być:

- użytkownik - domena (zbiór obiektów zależy od id użytkownika)
- proces - domena (zbiór obiektów zależy od id procesu)
- procedura - domena (zbiór obiektów zależy od lokalnych zmiennych procedury)

Realizacja domeny - UNIX

Domena związana z użytkownikiem

przełączanie domen - czasowa zmiana identyfikacji użytkownika:

dla każdego pliku mamy:

- ID właściciela
- bit domeny (*setuid bit*)

gdy bit domeny=0 - *user_id* procesu=*user_id* użytkownika, który uruchomił proces.

gdy bit domeny=1 - *user_id* procesu=*user_id* użytkownika, który jest właścicielem pliku.

Macierz dostępów

	F_1	F_2	F_3	drukarka
D_1	read		read	
D_2				print
D_3		exec	read	
D_4	read write		read write	

Macierz dostępów

	F_1	F_2	F_3	drukarka	D_1	D_2	D_3	D_4
D_1	read		read			przeł		
D_2				print			przeł	przeł
D_3		exec	read					
D_4	read write		read write			przeł		

Modyfikacja macierzy dostępów

Umożliwienie kontrolowanych zmian zawartości macierzy dostępów wymaga praw:

- kopiowania (dla obiektu)
- właściciela
- kontroli

Prawo kopiowania (*) – pozwala na skopiowanie prawa dostępu w obrębie kolumny:

- przekazanie (kopiowanie - utrata praw)
- ograniczone kopiowanie (tworzy prawo bez *)

Prawo kopiowania (*)

	F_1	F_2	F_3
D_1	read		read*
D_2			
D_3		execute	
D_4	read write		

Prawo właściciela (owner)

	F_1	F_2	F_3
D_1	read		read* owner
D_2			
D_3		execute	
D_4	read write		

Prawo kontroli (control)

Prawo kontroli jest stosowane tylko w odniesieniu do obiektów będących domenami.

proces działający w domenie D_2 może zmienić domenę D_4

	F_1	F_2	F_3	drukarka	D_1	D_2	D_3	D_4
D_1	read		read			przeł		
D_2				print			przeł	przeł control
D_3		exec	read					
D_4	read write		read write			przeł		

Implementacja macierzy dostępów

- **tablica globalna** - zbiór uporządkowanych trójek:

  <domena, obiekt, zbiór praw> $<D_i, O_j, R_k>$
  - zbyt duża tablica (pam. wirtualna)
  - brak możliwości grupowania obiektów lub domen
- **wykazy dostępów do obiektów**
  - dla każdej kolumny uporządkowane pary $<D_i, R_k>$ $R_k \neq \emptyset$
  - domyślny zbiór praw dostępu
- **wykazy uprawnień do domen** - spis $<O_j, R_k>$
  - obiekt chroniony, niedostępny dla procesu bezpośrednio
- **mechanizm zamka-klucza** dla każdego obiektu - wykaz wzorców binarnych (zamków), dla każdej domeny - wykaz wzorców binarnych (kluczy); gdy klucz pasuje do zamka proces może mieć dostęp do obiektu

Cofanie praw dostępu

- proste dla wykazu dostępów, natychmiastowe
- trudne dla wykazu uprawnień dla domen - uprawnienia rozproszone
  - **wtórne pozyskiwanie** - okresowe usuwanie uprawnień z każdej domeny.
  - **wskaźniki zwrotne** - dla każdego obiektu - wykaz wskaźników do uprawnień (kosztowny).
  - **adresowanie pośrednie** - uprawnienia wskazują na wpisy w tablicy globalnej wskazujące na obiekty (nie na obiekty bezpośrednio; cofnięcie uprawnienia - usunięcie elementu z tablicy globalnej).
  - **klucze** - zmiana klucza głównego związanego z danym obiektem - usunięcie wszystkich uprawnień danego obiektu.

Bezpieczeństwo

- ochrona - problem wewnętrzny
- bezpieczeństwo - wymaga systemu ochrony oraz zabezpieczenia przed środowiskiem zewnętrznym

  naruszenia bezpieczeństwa:
  - rozmyślne
  - przypadkowe

Uwierzytelnianie

Uwierzytelnianie to sprawdzanie tożsamości użytkownika

- klucz, karta (stan posiadania)
- nazwa użytkownika i hasło (wiedza)
- odcisk palca, podpis, wzorzec siatkówki oka (atrybut)

HASŁA

- odgadywanie haseł, metody siłowe próbowanie wszystkich możliwych kombinacji
- hasła - generowane przez system, wybierane
- postarzanie haseł
- szyfrowanie (UNIX)
- hasła jednorazowe (dobrane parami); algorytmiczne (ziarno, tajemnica)

Zagrożenia programowe

- koń trojański (segment kodu nadużywający swojego środowiska np. w edytorze)
  - zagrożenie - długie ścieżki dostępu
  - program naśladujący *login* przechwytujący hasło
- boczne wejście - pozostawienie luki w oprogramowaniu przez projektanta
  - obchodzenie procedur bezpieczeństwa dla pewnego użytkownika
  - oszustwa bankowe, kompilator

Zagrożenia systemowe

- robaki
  - mechanizm rozmnażania - paraliżowanie działania systemu
  - 1988 rok, Morris - robak internetowy - program haczący, program główny (rsh, finger, sendmail)
- wirusy - rozchodzą się po innych programach siejąc spustoszenie
  - fragment kodu osadzony w poprawnym programie

Polepszenie bezpieczeństwa

- nadzorowanie zagrożeń - śledzenie podejrzanych zachowań (zliczanie błędów logowania)
- dziennik kontroli (zapis czasu, nazwy użytkownika, dostępów)
- analizowanie systemu okresowo
  - krótkie, łatwe hasła
  - manipulowanie **setuid**
  - działalność nieupoważnionych programów w katalogach systemowych
  - długo liczące się procesy
  - ochrona katalogów i plików
  - wpisy w PATH

Bezpieczne łączenie komputerów z niepewną siecią

Zapora ogniowa (firewall)

- Oddzielenie systemów
- Komputer lub ruter ograniczający dostęp sieciowy
- nadzoruje i rejestruje wszystkie połączenia
  - domena niepewna (Internet)
  - strefa zdemilitaryzowana (o ograniczonym zaufaniu) - DMZ
  - trzecia domena - komputery zainstalowane w firmie

dopuszczanie ruchu tylko na wybranych portach;

system: deny all; allow all

System Linux

- zapora oparta na jądrze - szeroki zakres funkcji zabezpieczających
- zapory na poziomie aplikacji (gateway, proxy)

  np. serwer proxy - odbiera żądanie od klientów (z zewnątrz i z wewnątrz) - działa w oparciu o zdefiniowane reguły - dopuszcza lub nie pewne działania

  ignorowanie wiadomości e-mail o pewnym rozmiarze

Szyfrowanie

E - algorytm szyfrowania

D - algorytm deszyfrowania

k - tajny klucz dostarczany z aplikacją

m - komunikat

STANDARD SZYFROWANIA DANYCH (DES)

- $D_k(E_k(m))=m$.
- D_k oraz E_k - można wykonać efektywnie
- bezpieczeństwo systemu zależy jedynie od tajności klucza

Szyfrowanie kluczem jawnym

klucz jawny: (e,n)

klucz prywatny: (d,n)

e,d,n - dodatnie liczby całkowite

m - komunikat, reprezentowany jako liczba <1;n-1>

$$E(m)=m^e \bmod n=C$$

$$D(C)=C^d \bmod n$$

n - jawne, d, e trudne do odgadnięcia

Szyfrowanie kluczem jawnym

$n=p \cdot q$

p,q - losowo wybrane liczby pierwsze (min. 100-cyfrowe)

d: $NWD(d, (p-1) \cdot (q-1))=1$

e: $(e \cdot d) \bmod ((p-1) \cdot (q-1))=1$

np.

$p=5; q=7; n=35;$

$(p-1) \cdot (q-1)=24; \quad d=11; \quad e=11$

dla $m=3$

$C=E(m)=m^e \bmod n=3^{11} \bmod 35=12$

$D(C)=C^d \bmod n= 12^{11} \bmod 35=3=m$

Klasyfikacja poziomów bezpieczeństwa

- Cztery klasy bezpieczeństwa: A,B,C,D
- D - systemy nie spełniające wymagań żadnej z innych klas (MS-DOS, Windows 3.1)
- C - umożliwienie wglądu w poczynania użytkowników (audit); dowolne reguły ochrony i odpowiedzialności; poziomy C1, C2
  - C1 -zawiera środki kontroli umożliwiające użytkownikom ochronę informacji (UNIX)
  - C2 - dodatkowo indywidualny poziom dostępu (np. prawa do pliku można określić w odniesieniu do poszczególnych osób) (niektóre systemy UNIX, NT)

Klasyfikacja poziomów bezpieczeństwa

- B - (B1, B2, B3) dla każdego obiektu - kategoria uwrażliwienia np. użytkownicy różnych poziomów (tajny, poufny); izolacja procesów - rozłączne przestrzenie adresowe
- A - funkcjonalnie równoważny B3; stosuje się formalna specyfikację projektu oraz techniki weryfikacji gwarantujące wysoki poziom pewności poprawnej implementacjii bazy bezpieczeństwa komputerowego (TCB)

TCB (*Trusted Computer Base*)

Zespół wszystkich systemów ochrony

- sprzęt
- oprogramowanie
- oprogramowanie układowe
  - gwarantuje zdolność egzekwowania przez system zakładanej polityki bezpieczeństwa
  - nie określa polityki (zasady bezpieczeństwa związane z uzyskaniem certyfikatu)