

Systemy operacyjne

WYKŁAD 13 i 14

dr inż. Stanisława Plichta
splichta@ans-ns.edu.pl

Przetwarzanie współbieżne, równoległe i rozproszone

- Przetwarzanie współbieżne, równoległe i rozproszone to powszechne dziś formy wykonania programów.
- O wykonaniu współbieżnym dwóch wątków mówimy wtedy, gdy rozkazy jednego wątku zaczną być wykonywane, zanim zakończy się wykonywanie rozkazów drugiego, uruchomionego wcześniej.
- O przetwarzaniu równoległym mówimy wtedy, kiedy przynajmniej niektóre z rozkazów wątków wykonywanych współbieżnie są realizowane w tym samym czasie.

Przetwarzanie równoległe

- Dwa lub więcej procesów (wątków) jednocześnie współpracuje w celu rozwiązania pojedynczego zadania.
- Obliczenia równoległe są silnie związane z dziedziną obliczeń wysokiej wydajności oraz obliczeniami naukowo-technicznymi.
- W czasach procesorów wielordzeniowych, programowanie równoległe jest koniecznością.

Rodzaje przetwarzania równoległego

Równoległość na poziomie:

- Pojedynczego rozkazu.
- Sekwencji rozkazów tworzących zadanie:
 - rozdział kolejno wykonywanych rozkazów pomiędzy procesory/rdzenie
 - podział zadania obliczeniowego na podzadania, przydział poszczególnych zadań procesorom/rdzeniom
- Programów.

Tworzenie programów równoległych

- W procesie tworzenia programów równoległych znaczenie ma:
 - Wykrycie dostępnej współbieżności w trakcie realizacji programu.
 - Określenie koniecznej synchronizacji lub wymiany komunikatów pomiędzy procesami lub wątkami realizującymi program.

Jednym z najważniejszych wymagań stawianych programom równoległym jest przenośność.

Przetwarzanie równoległe i rozproszone

- Istnieje kilka podstawowych celów, dla których stosuje się przetwarzanie równoległe i rozproszone:
 - Zwiększenie wydajności obliczeń,
 - Zwiększenie niezawodności przetwarzania.
 - Zwiększenie elastyczności wykorzystania dostępnych zasobów komputerowych.
- Do upowszechnienia sprzętu równoległego przyczyniło się wprowadzenie klastrów.

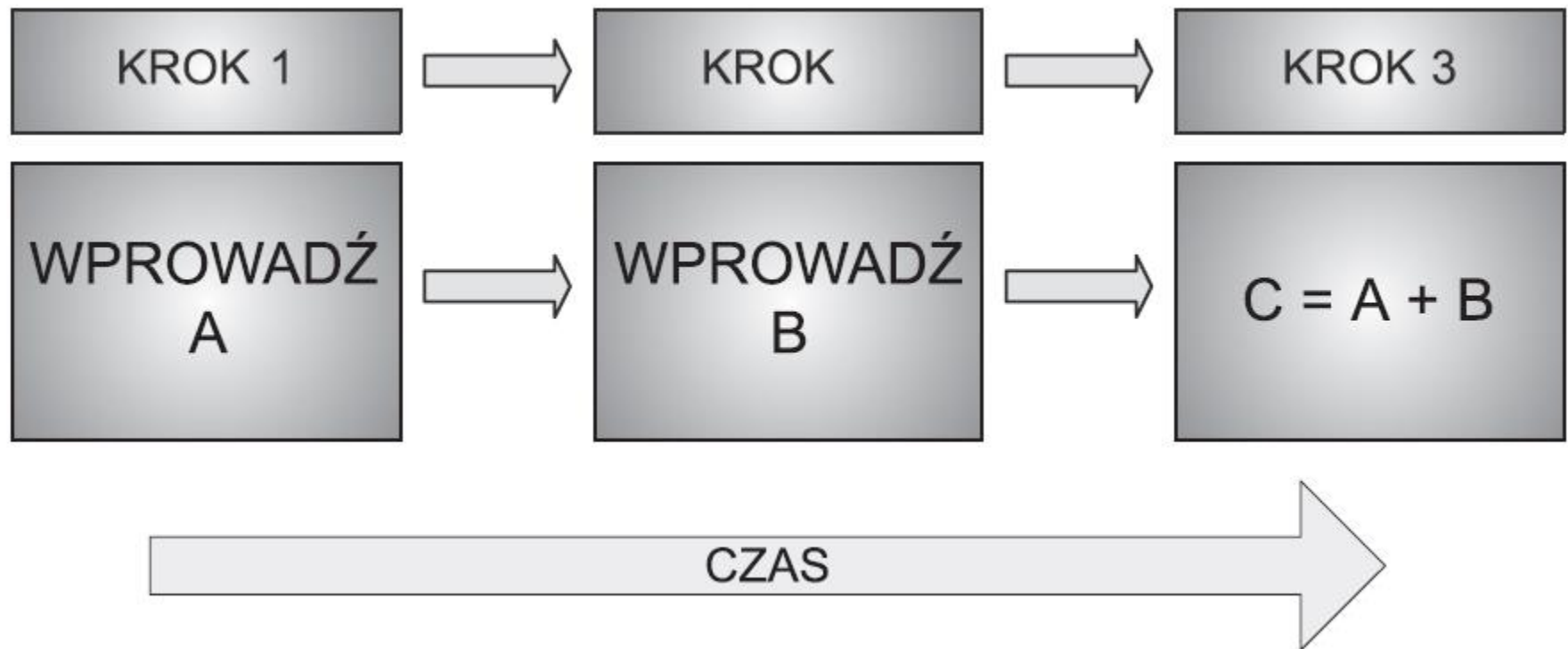
KLASYFIKACJA SYSTEMÓW RÓWNOLEGŁYCH

Klasyfikacja Flynna:

- W klasyfikacji Michaela Flynna podstawą do rozróżniania poszczególnych rodzajów maszyn obliczeniowych jest zdolność obsługi strumienia danych i strumienia rozkazów.
- Maszyny równoległe zostały w niej podzielone na 4 grupy:
 - SISD (*Single Instruction stream, Single Data stream*)
 - SIMD (*Single Instruction stream, Multiple Data stream*)
 - MIMD (*Multiple Instruction stream, Multiple Data stream*)
 - MISD (*Multiple Instruction stream, Single Data stream*)

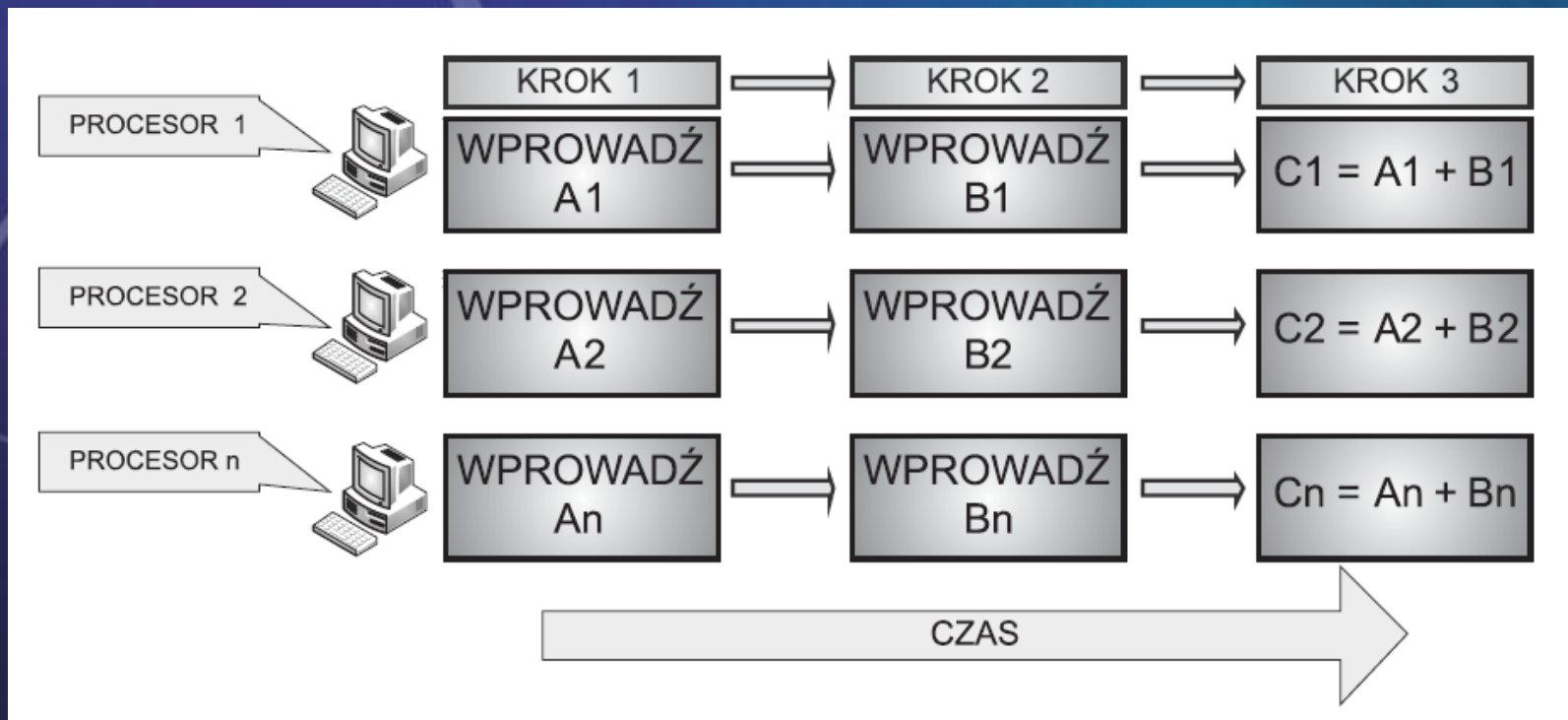
Architektura SISD

Architektura komputera sekwencyjnego jednoprocessorowego, zwana też architekturą Neumanna.



Architektura SIMD

- Metoda zastosowana we wczesnych systemach równoległych.
- Każdy z procesorów przetwarza ten sam zestaw instrukcji na własnym zestawie danych.

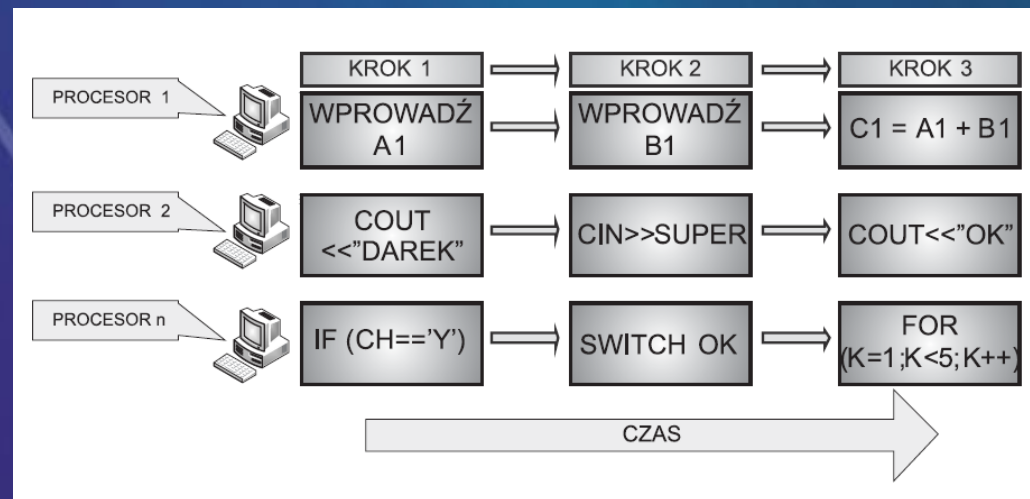


Architektura MIMD

- Każdy z procesorów pracuje z własnym zestawem instrukcji, operując na własnym zestawie danych.
- Praca systemu opartego na architekturze MIMD ma charakter asynchroniczny
- Procesory pracują całkowicie oddzielnie, realizując oddzielne i najczęściej różne strumienie rozkazów - procesy.

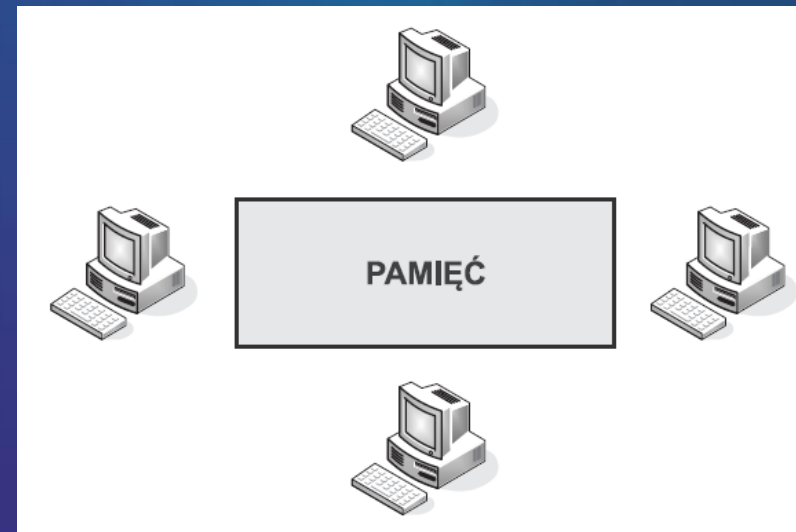
W ramach grupy MIMD wyróżnia się:

- MIMD z pamięcią wspólną,
- MIMD z pamięcią rozproszoną,
- MIMD z pamięcią rozproszono-wspólną



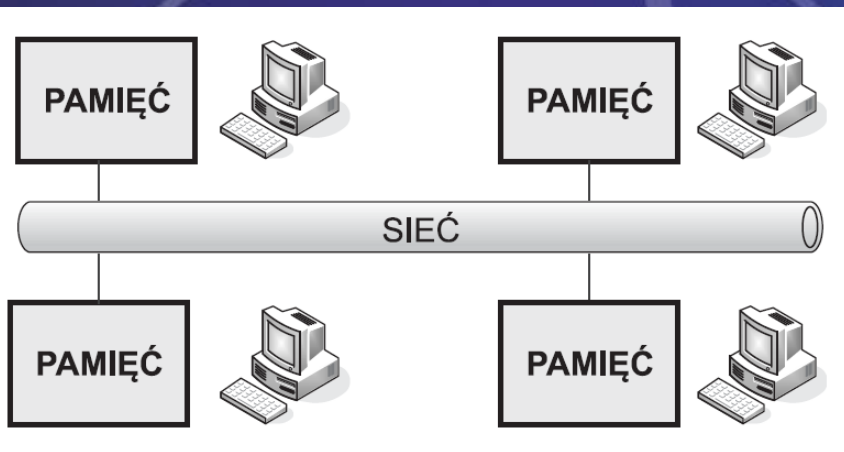
Architektura MIMD-SM

- Procesory są połączone specjalizowaną siecią interconnect, poprzez którą komunikują się ze wspólnym obszarem pamięci.
- W ramach systemów SM można wyróżnić dwie kolejne kategorie:
 - **Shared everything**
 - **Shared something**

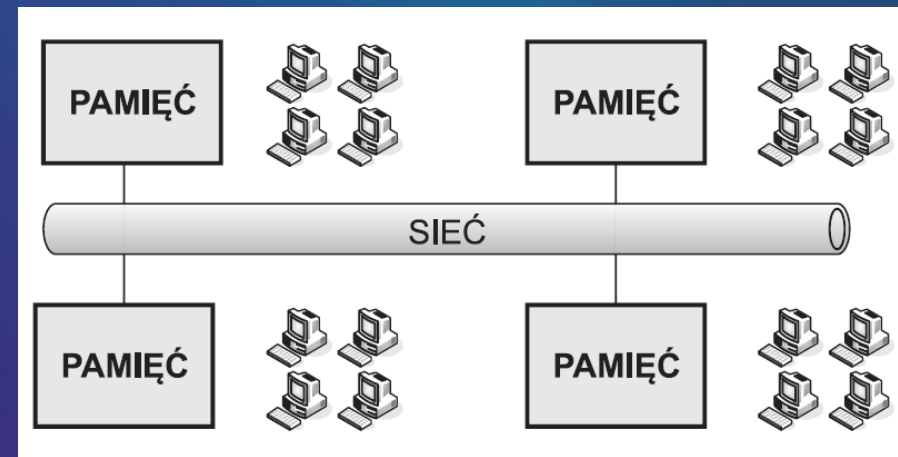


Architektura MIMD

- Każdy procesor posiada własną pamięć, dostępną tylko dla niego samego.
- Procesory przekazują sobie nawzajem informacje poprzez komunikaty.
- Są to zazwyczaj maszyny wieloprocessorowe.



Architektura MIMD-DM



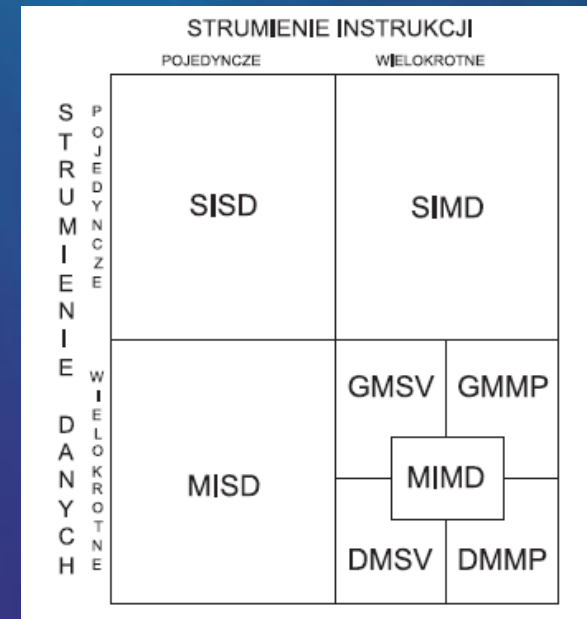
Architektura MIMD-HDSM

Architektura MISD

- Maszyny tego typu wykonują różne operacje na tych samych danych.
- Zbudowano niewiele takich maszyn, nie są one używane do celów komercyjnych.
- Znajdują zastosowanie dla uzyskania bardzo wysokiej niezawodności, np. sterowanie reaktorami jądrowymi.

Rozszerzona klasyfikacja Johnsona

- Większość systemów przeznaczonych do obliczeń równoległych jest klasyfikowana jako maszyny MIMD - Johnson usystematyzował tę grupę, wprowadzając jako kryterium podziału strukturę pamięci.
- Według Johnsona w ramach architektury MIMD możemy wyróżnić 4 grupy:
- **GMSV** (ang. *Global Memory Shared Variables*)
- **GMMP** (ang. *Global Memory Message Passing*)
- **DMSV** (ang. *Distributed Memory Shared Variables*)
- **DMMP** (ang. *Distributed Memory Message Passing*)

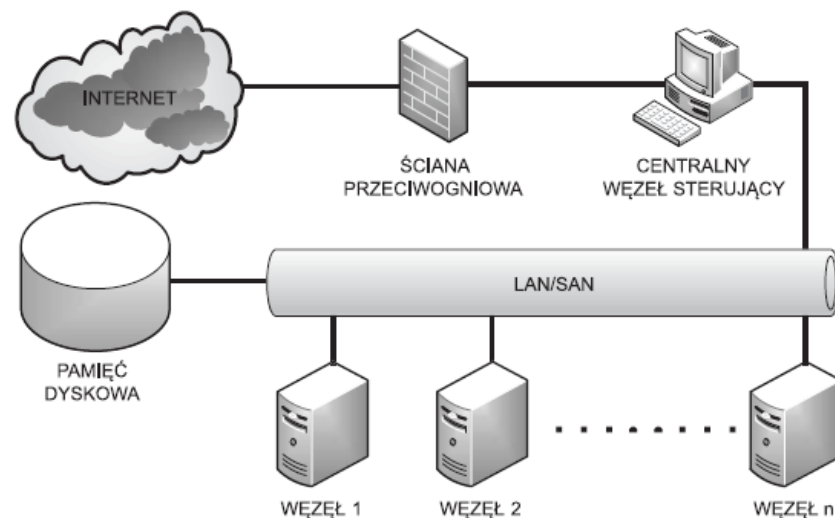


Klasyfikacja Tanenbauma

- Odmienne spojrzenie na klasę MIMD przedstawił Andrew S. Tanenbaum – koncentracja na sposobie w jaki komunikują się poszczególne systemy.
- Stopień rozproszenia systemów określany jest na podstawie szybkości przesyłania danych pomiędzy systemami.
 - systemy ściśle powiązane
 - systemy słabo powiązane

Ogólny model klastra komputerowego

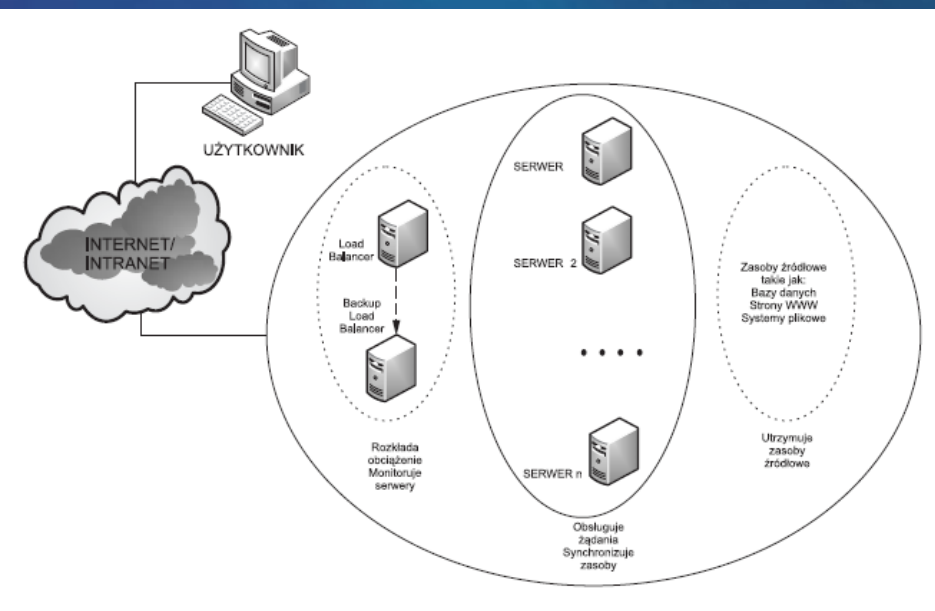
- Klaster komputerowy to grupa wspólnie działających połączonych maszyn, które tworzą pojedynczy zespół obliczeniowy.
- Klastry składają się z węzłów (*node*), czyli pojedynczych maszyn, które są połączone za pomocą wydajnej sieci – zwykle szybkiej sieci lokalnej.
- Ilość węzłów w klastrze może wahać się od kilku do kilku tysięcy.
- Architektura klastra zależy od danej implementacji.



Klasy równoważące obciążenia (*Load Balancing Cluster*)

- Służą do utrzymywania w działaniu mocno obciążonych usług sieciowych takich jak np. serwery WWW, czy serwery bazy danych.
- Działanie ich polega na równoważnym dystrybuowaniu obciążenia pomiędzy poszczególne węzły klastra.

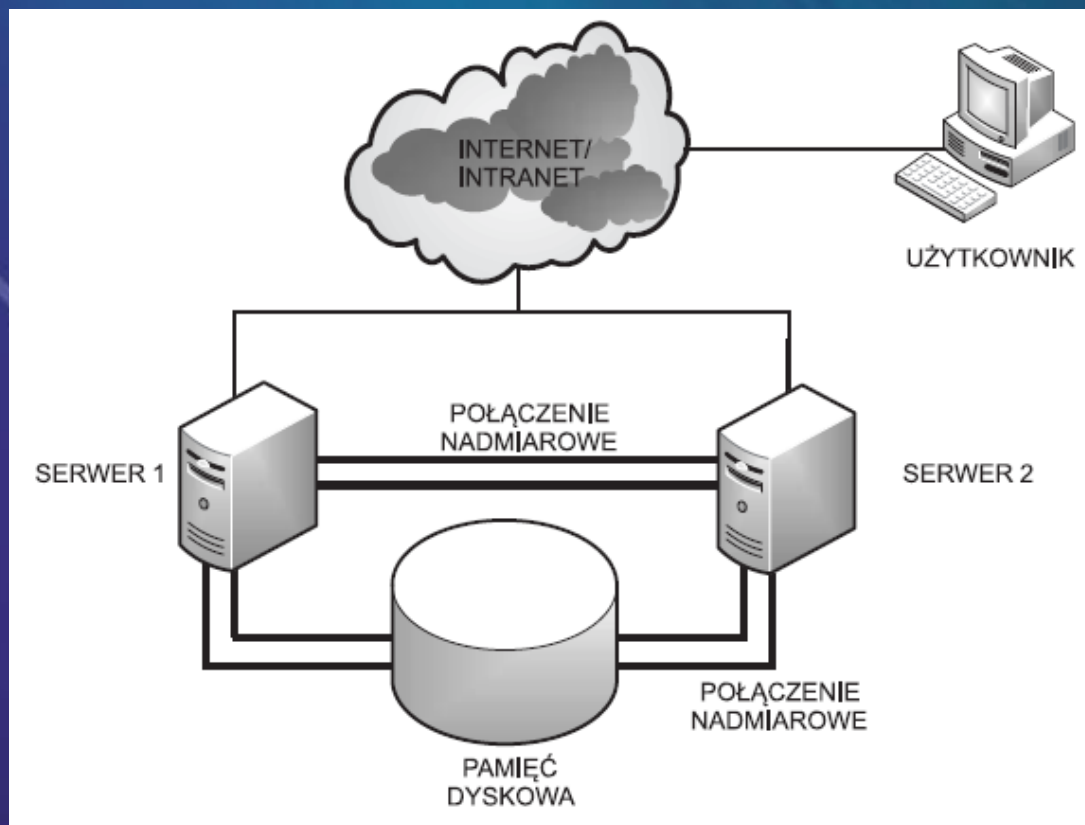
Klasy tego typu implementuje się w przypadku, gdy bardzo istotny jest czas reakcji usługi na żądania klienta.



Klasy niezawodnościowe (*High Availability Clusters*)

Zadaniem tych klastrów nie jest zwiększanie wydajności ale wyeliminowanie tak zwanego pojedynczego punktu awarii.

Działanie ich polega na rozłożeniu w przypadku wystąpienia awarii któregoś z serwerów jego zadań na pozostałe serwery, w taki sposób, aby nie było to widoczne dla użytkowników systemu.



Klastry wydajnościowe (*High Performance Clusters*)

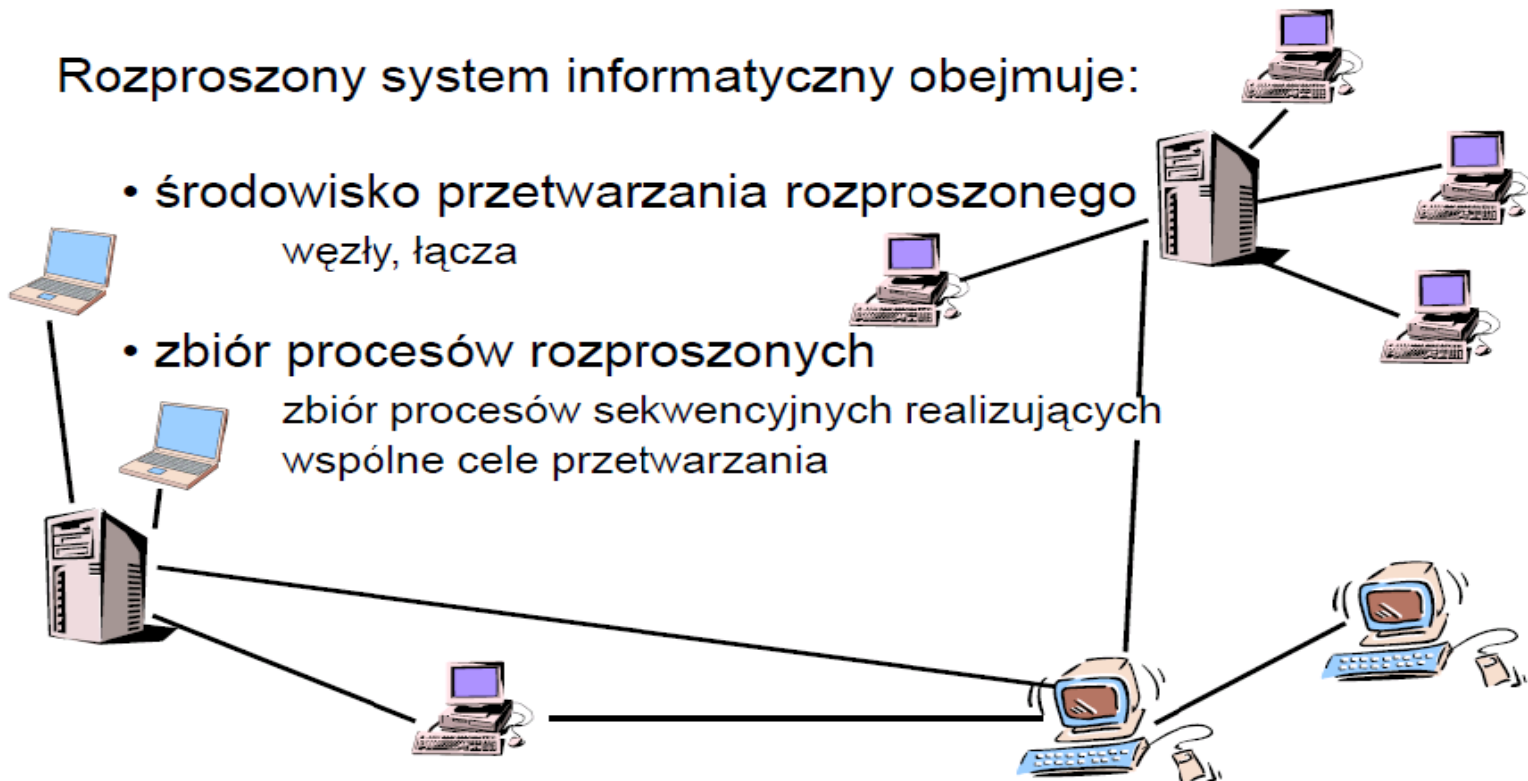
- Używane są do masowego przetwarzania danych jednego rodzaju, przy czym równoważenie obciążenia najczęściej leży w gestii samych aplikacji, jednak nie jest to regułą.
- W klastrach tych duże znaczenie ma aplikacja, którą najczęściej należy przygotować w jednej z dostępnych często specjalizowanych bibliotek programistycznych.
- Zazwyczaj od aplikacji zależy w jakim stopniu będzie możliwe odpowiednie rozłożenie obciążenia, a przez to przyśpieszenie obliczeń.

Rozproszony system informatyczny

Rozproszony system informatyczny

Rozproszony system informatyczny obejmuje:

- środowisko przetwarzania rozproszonego
węzły, łącza
- zbiór procesów rozproszonych
zbiór procesów sekwencyjnych realizujących
wspólne cele przetwarzania



Systemy rozproszone - wymagania

Wymagania projektowe systemów rozproszonych:

- Współdzielenie zasobów
- Otwartość
- Współbieżność
- Skalowalność
- Wydajność
- Tolerowanie uszkodzeń (niezawodność)
- Przezroczystość

Dzielenie zasobów

Zasoby:

- **Składowe sprzętowe:** specjalizowane komputery, dyski, drukarki, urządzenia peryferyjne i inne.
- **Składowe programowe:** usługi sieciowe, aplikacje internetowe.
- **Zasoby informacyjne:** pliki, bazy danych.

System rozproszony powinien umożliwiać wspólne wykorzystanie zasobów lokalnych, znajdujących się w węzłach systemu rozproszonego, przez wielu użytkowników zdalnych.

Dzielenie zasobów

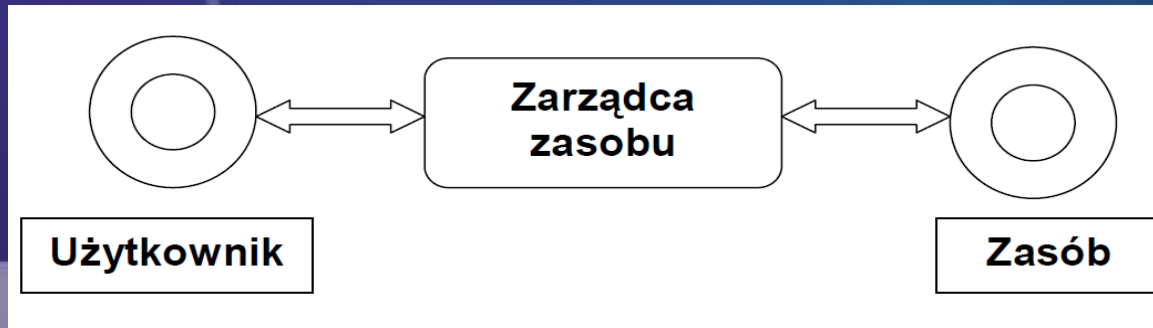
Cele współdzielenia zasobów:

- poprawia efektywność wykorzystania zasobów,
- zmniejsza koszt budowy systemu,
- zwiększa efektywność przetwarzania systemu,
- zwiększa dostępność systemu,
- umożliwia pracę zespołową i zdalne nauczanie.

Współdzielenie zasobów nie powinno naruszać zasad polityki bezpieczeństwa systemu rozproszonego w zakresie: poufności, integralności i dostępności informacji.

Koncepcja dostępu do zasobów

- Zasoby dostępne fizycznie w węzłach.
- Konieczny interfejs komunikacyjny programowo-sprzętowy, który umożliwi użytkownikowi korzystanie z zasobu.
- Obsługa dostępu do zasobów realizowana przez oprogramowanie zarządcy zasobu.
- **Zarządca zasobu:** moduł oprogramowania (proces) zarządzający zasobem określonego typu.
- **Użytkownik zasobu:** moduł oprogramowania (proces) żądający dostępu do zasobu.



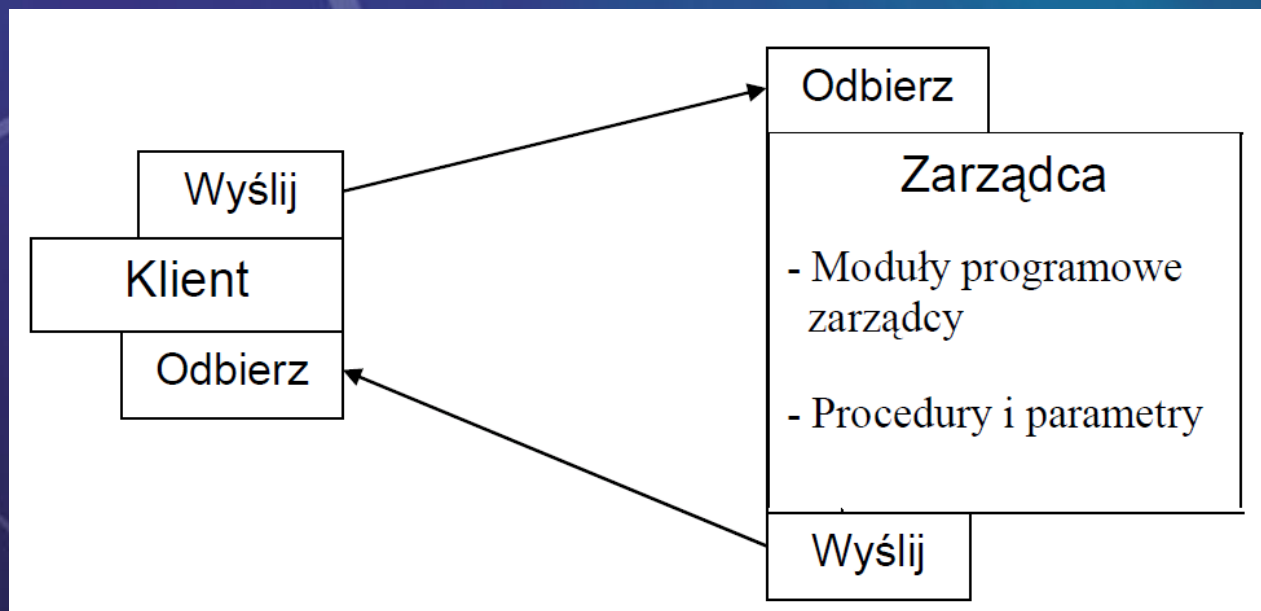
Koncepcja dostępu do zasobów

Zadania zarządcy zasobu:

- Zapewnienie jednolitego schematu nazewniczego dla każdej klasy zasobów.
- Umożliwienie dostępu do poszczególnych zasobów z dowolnego miejsca.
- Odwzorowywanie nazw zasobów na adresy komunikacyjne.
- Synchronizacja współbieżnego dostępu do zasobów.
- Zapewnienie zgodności stanu zasobu z rzeczywistością.

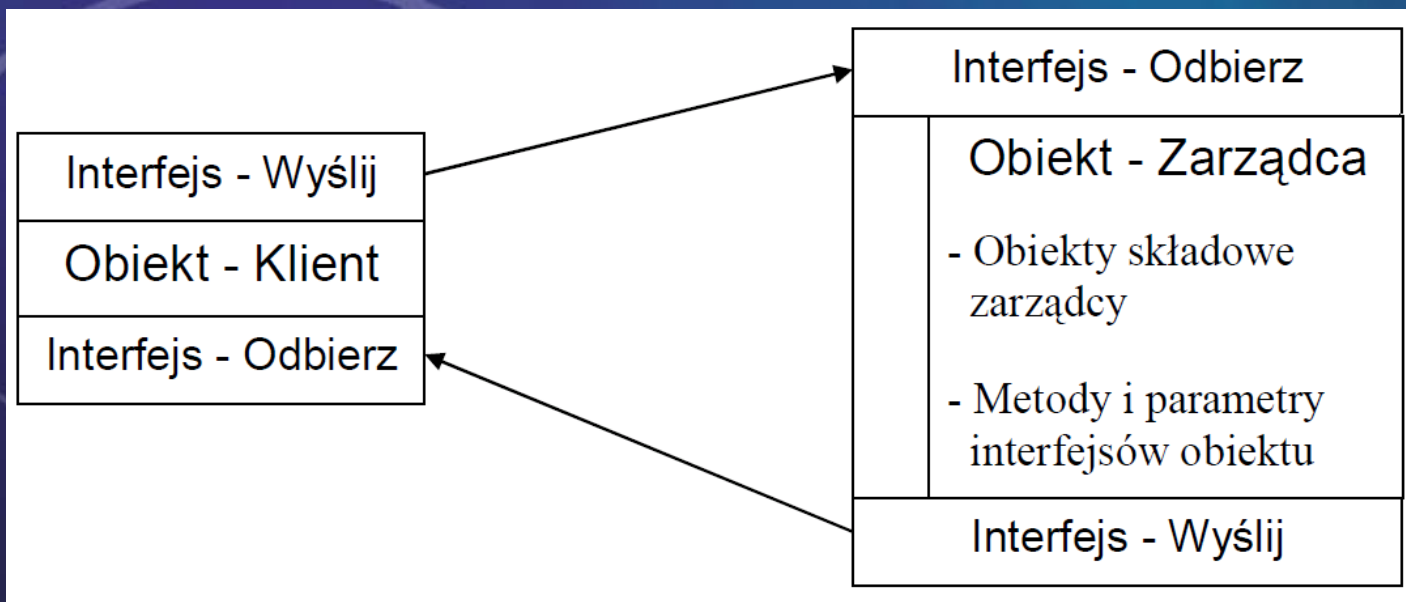
Modele współpracy użytkownika z zarządcą

Model proceduralno-komunikatowy - proces klienta i serwera współpracują ze sobą w oparciu o przesyłanie komunikatów.



Modele współpracy użytkownika z zarządcą

Model obiektowy - procesy klienta i zarządcy są obiektami o jednoznacznych identyfikatorach, które posiadają interfejsy, zawierające metody umożliwiające wykonywanie operacji z udziałem obiektów, a także dostęp do danych i zasobów.



Popularne architektury rozproszenia

- Klient-serwer
- Klient-multi-serwer
- *peer-to-peer (P2P)*
- Architektura oparta na oprogramowaniu pośredniczącym (*middleware*)

Podstawowe pojęcia

- Usługa
- System obsługi (serwer)
- Klient
- Nazewnictwo
- Przezroczystość lokalizacji
- Niezależność lokalizacji

Pamięć podręczna

- Pamięć podręczna w rozproszonych systemach plików jest wykorzystywana w celu zmniejszenia ruchu w sieci.
- Jeśli dane potrzebne do wykonania zamówienia nie znajdują się w pamięci podręcznej – przenosi się ich kopię z systemu obsługi do użytkownika.
- Problem spójności pamięci podręcznej.
- Ziarnistość pamięci podręcznej.
- Niezawodność pamięci podręcznej.

Aktualizacja kopii głównej

- Metoda ustawicznego przepisywania.
- Metoda opóźniania aktualizowania kopii głównej:
 - Bloki danych mogą być przesyłane tylko wtedy gdy przewidywane jest usunięcie bloku z pamięci podręcznej.
 - Bloki danych są zapisywane w chwili zamykania pliku – daje to korzyści przy plikach otwieranych na długie okresy czasu i często modyfikowanych.

Rozproszony system plików

Zasady jednoczesnego dostępu wielu użytkowników do pliku:

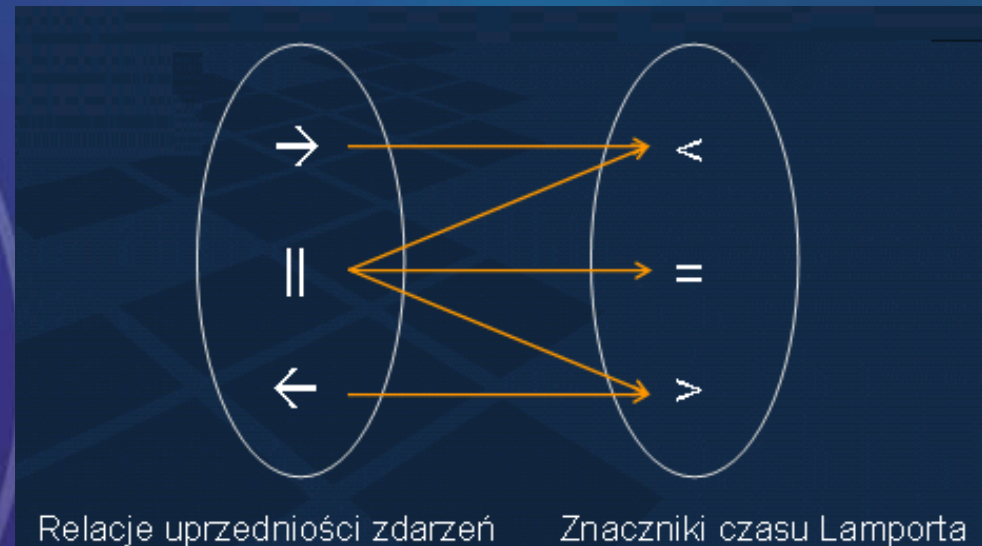
- **Możliwość jednoczesnego odczytu bez możliwości zapisywania** – mechanizm prosty do realizacji.
- **Kontrolowanie zapisu** - wielu użytkowników otwiera jeden plik, ale tylko jeden z nich ma możliwość zapisu zmienionego dokumentu.
- **Jednoczesne zapisywanie** - daje możliwość jednoczesnego zapisu i odczytu tego samego zbioru przez kilku użytkowników - wymaga intensywnego nadzoru ze strony systemu operacyjnego.
 - klasa *stateless*
 - klasa *callback*

Synchronizacja w systemach rozproszonych

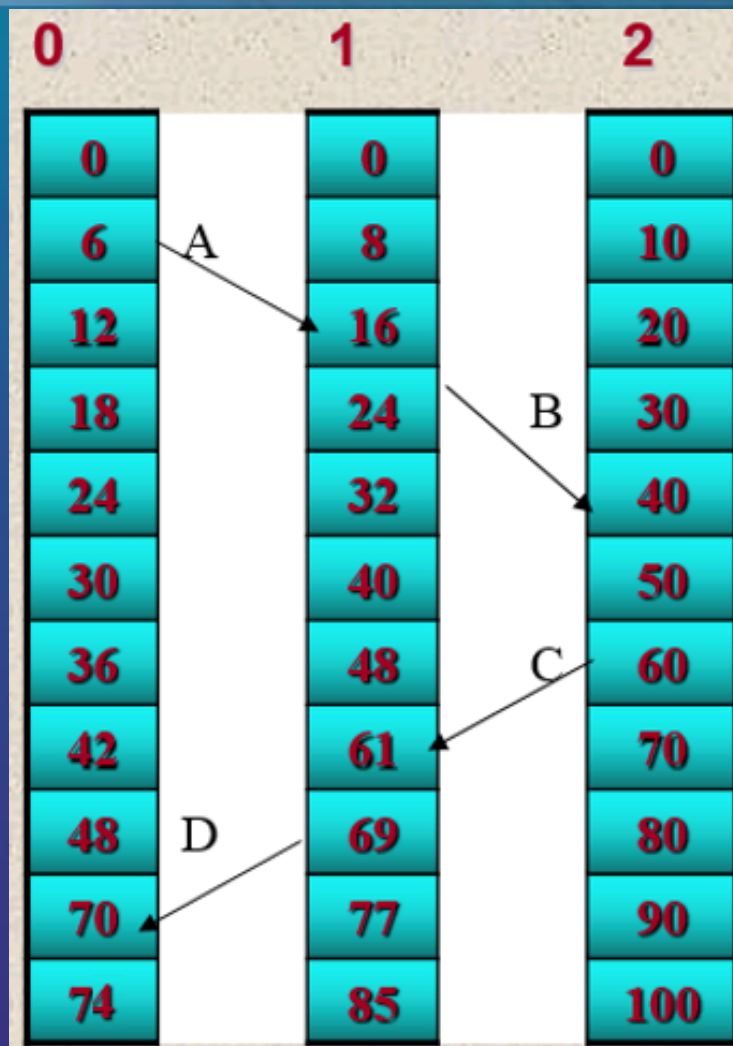
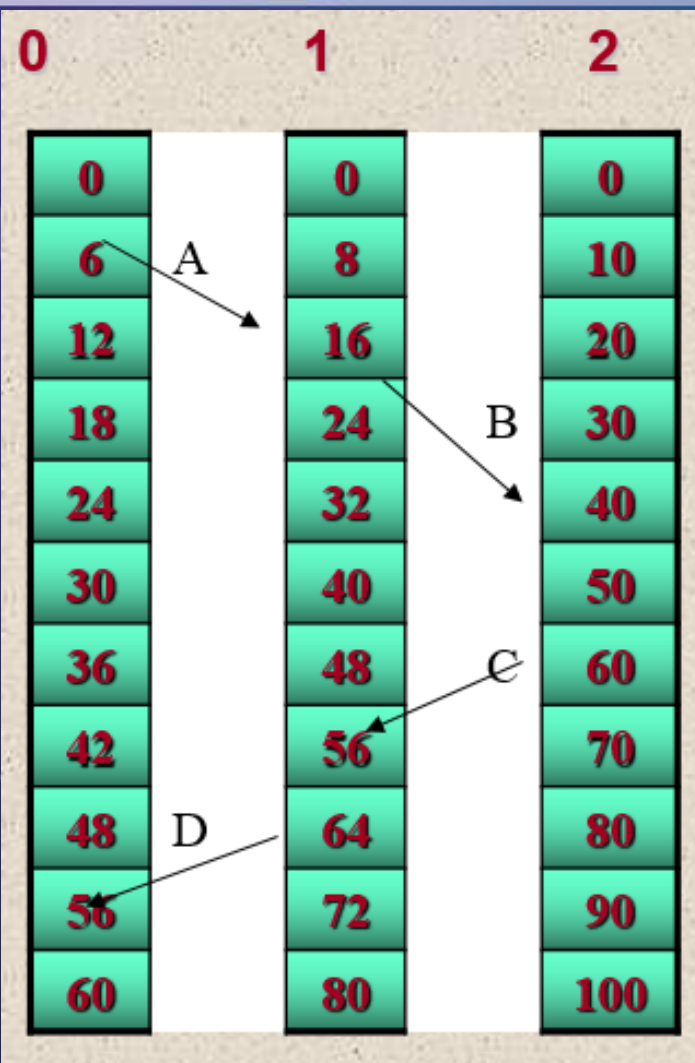
- Synchronizacja w systemach rozproszonych jest procesem bardziej złożonym niż w systemach scentralizowanych.
- W zagadnieniach synchronizacji procesów na wielu maszynach nie jest istotna znajomość czasu absolutnego - istotne jest, aby procesy mogły ustalić kolejność zdarzeń.
- Powodu pomiar czasu możemy podzielić na:
 - fizyczny czasu pomiar – zegary wskazują czas astronomiczny,
 - Logiczny pomiar czasu – zegary o wzajemnie uzgodnionym czasie, niekoniecznie astronomicznym.

Algorytm Lamporta

- Jeśli a poprzedza b w tym samym procesie, to $C(a) < C(b)$
- Jeśli a i b oznaczają nadanie i odbiór komunikatu, to $C(a) < C(b)$
- Dla wszystkich zdarzeń a i b $C(a) \neq C(b)$

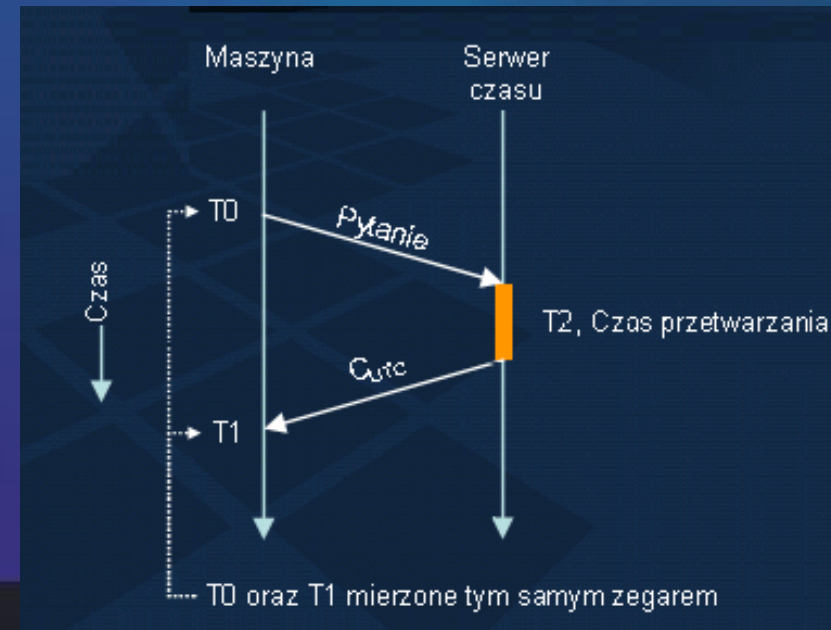


Zegary logiczne



Zegary fizyczne – algorytm Cristiana

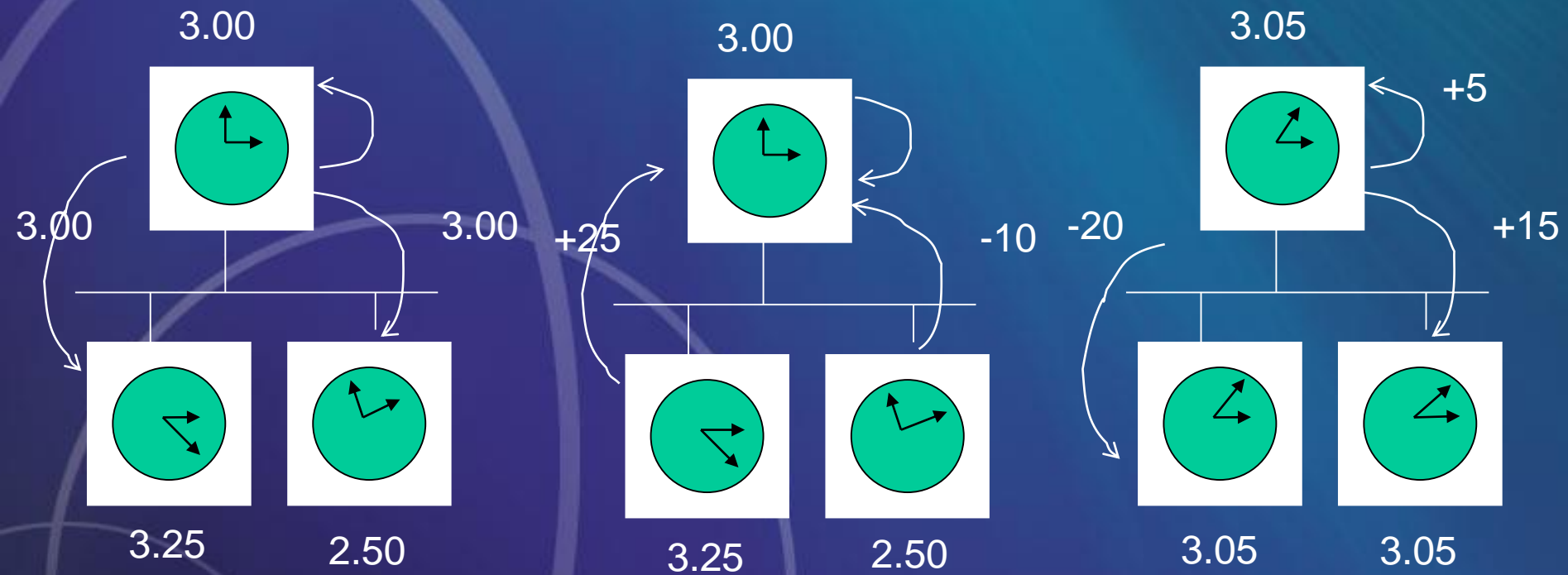
- UTC – wzorzec uniwersalnego koordynowanego czasu.
- Czas UTC jest udostępniany przez kilka satelitów.
- Serwer czasu w algorytmie Cristiana jest pasywny - inne maszyny okresowo zadają mu pytania o czas - jego działanie sprowadza się do udzielania im odpowiedzi.



Zegary fizyczne – algorytm Berkeley

- W systemie Berkeley przyjęto dokładnie przeciwne założenie - serwer czasu jest aktywny i odpytuje okresowo każdą maszynę o jej aktualny czas.
- Na podstawie tych odpowiedzi oblicza czas średni i nakazuje wszystkim innym maszynom, przesunąć zegary na ten nowy czas lub je zwolnić, aż do osiągnięcia określonej redukcji czasu.

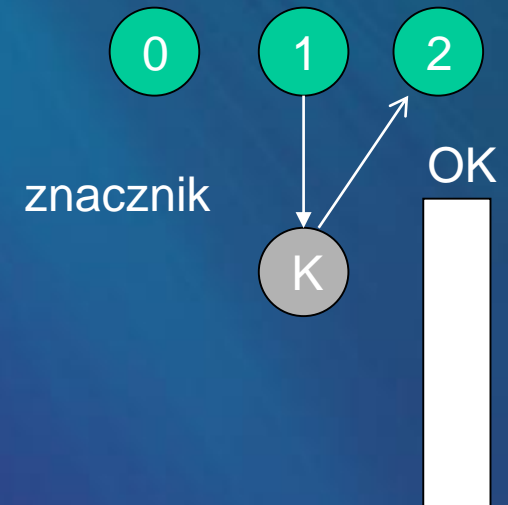
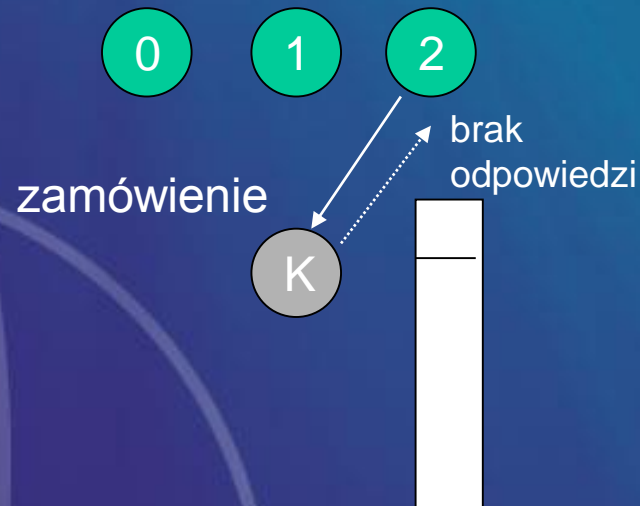
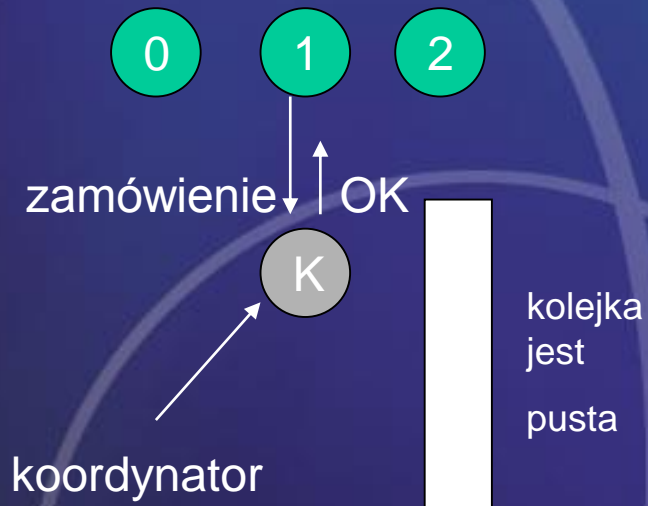
Algorytm uśredniania



Wzajemne wyłączanie w systemach rozproszonych

- Algorytm scentralizowany
- Algorytm rozproszony
- Algorytm pierścienia z żetonem

Algorytm scentralizowany



Algorytm scentralizowany

- W systemie istnieje proces będący koordynatorem zasobów.
- Wymienia on z procesami następujące typy komunikatów:

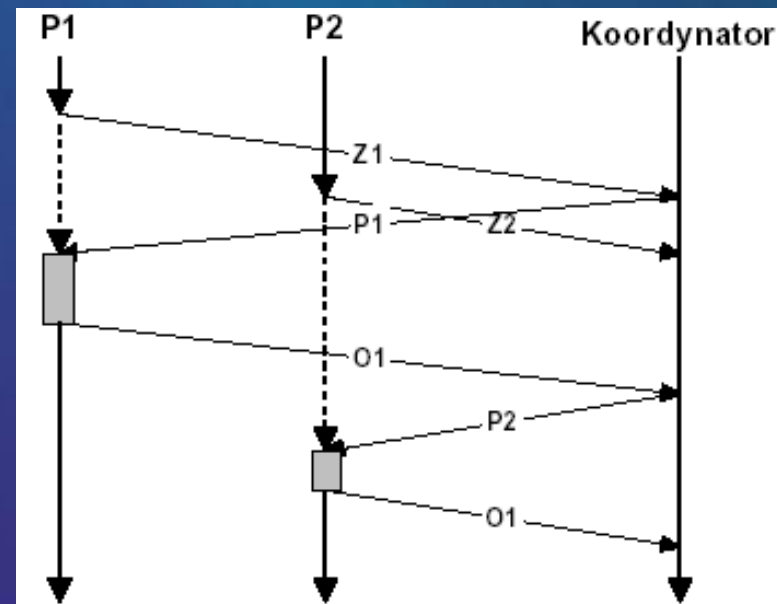
Z- Zamówienie zasobu Proces -> Koordynator

P- Przydzielenie zasobu Koordynator -> Proces

O- Oddanie zasobu Proces -> Koordynator

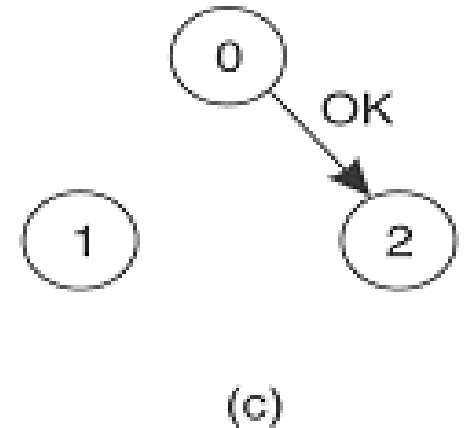
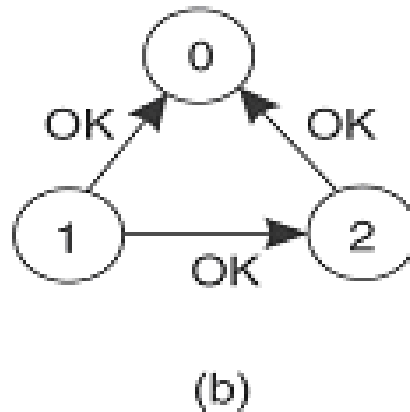
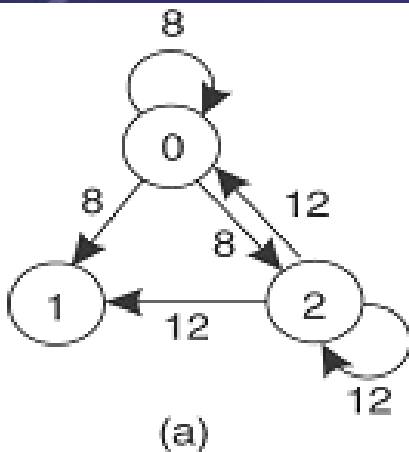
Proces działa według algorytmu:

1. Wysyła do koordynatora zamówienie zasobu.
2. Czeki na przydzielenie zasobu.
3. Używa zasobu.
4. Wysyła do koordynatora komunikat o zwolnieniu zasobu.

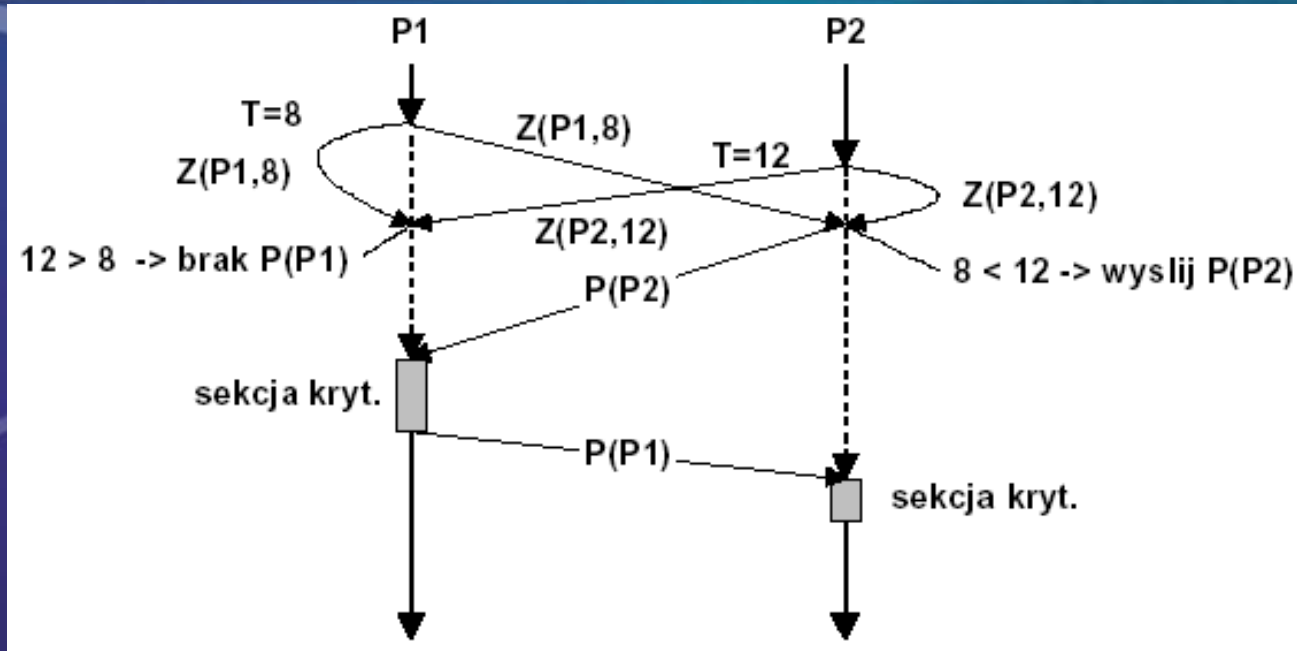


Algorytm rozproszony

- Proces, który chce wejść do sekcji krytycznej, buduje komunikat z nazwą tej sekcji, swoim numerem i bieżącym czasem.
- Wysyła komunikat do wszystkich innych procesów.
- Każdy komunikat jest potwierdzany.
- Jeśli proces otrzyma zamówienie od innego procesu, to działanie jakie podejmie, będzie zależać od jego zamiarów związanych z sekcją krytyczną wymienioną w komunikacie.



Algorytm rozproszony



Własności:

- Algorytm poprawnie realizuje rozproszone wzajemne wykluczanie.
- Nie występuje zagłódzenie, gdyż zamówienia są uporządkowane czasowo.
- Awaria jednego procesu powoduje awarie całego systemu.
- Proces musi być gotowy na odbiór zamówień i odpowiedzi.

Algorytm rozproszony

Algorytm charakteryzuje się następującymi cechami:

- Zapewnia wzajemne wykluczanie w systemie rozproszonym.
- Wymaga, aby w systemie występowało całkowite uporządkowanie zdarzeń - zegary logiczne.

Rodzaje komunikatów:

- Zamówienie
- Pozwolenie

W komunikacie wysyła się:

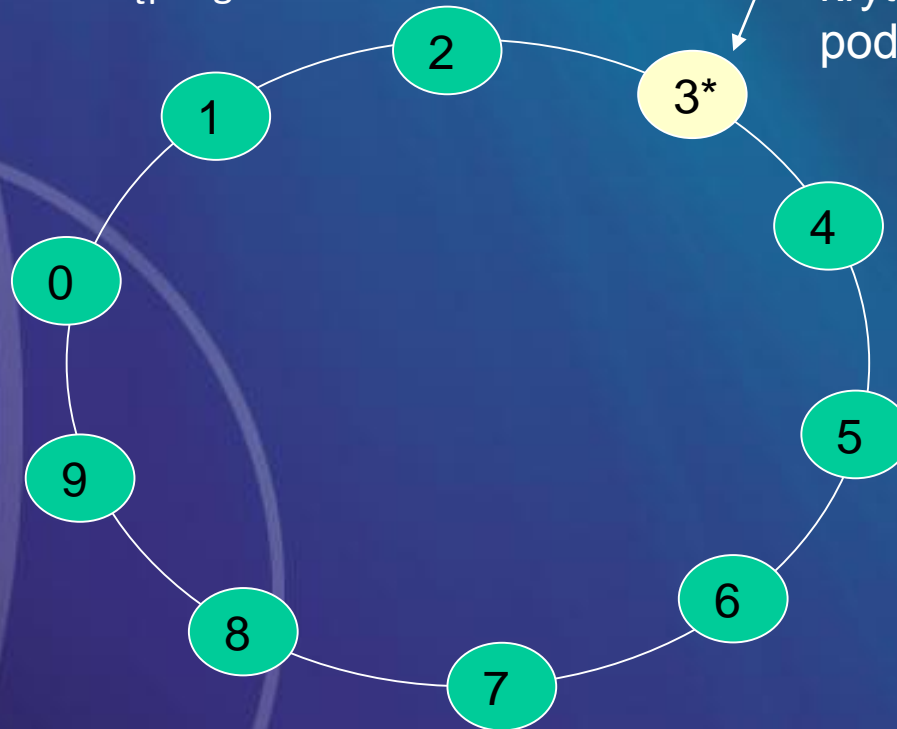
- Nazwę zasobu
- Identyfikator procesu
- Wartość zegara logicznego

Algorytm pierścienia z żetonem

Gdy proces chce wejść do sekcji krytycznej:

- Oczekuje na żeton.
- Gdy otrzyma żeton wchodzi do sekcji.
- Po wyjściu przekazuje żeton do następnego procesu w pierścieniu.

posiadacz żetonu
wchodzi do sekcji
krytycznej lub
podaje żeton dalej



Gdy proces nie chce wejść do sekcji krytycznej przekazuje żeton następnemu procesowi.

Porównanie algorytmów

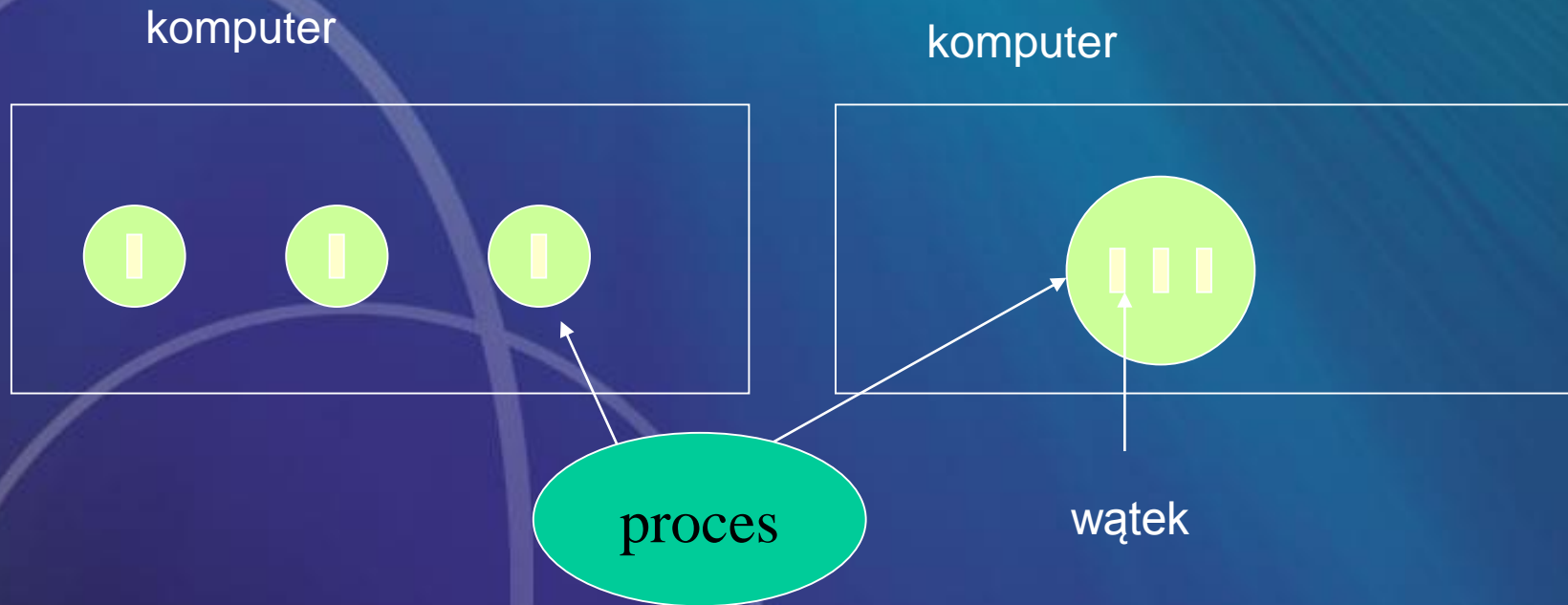
Algorytm	Liczba komunikatów na we/wy	Opóźnienie wejścia (mierzone liczbą komunikatów)	Problemy
Scentralizowany	3	2	Awaria koordynatora
Rozproszony	$2(n-1)$	$2(n-1)$	Awaria dowolnego procesu
Pierścienia z żetonem	od 1 do ∞	od 0 do $n-1$	Utrata żetonu awaria procesu

Procesy rozproszone

- **Proces rozproszony** - współbieżne i skoordynowane wykonywanie zbioru procesów sekwencyjnych w środowisku rozproszonym, które współdziałają, aby osiągnąć wspólny cel przetwarzania.
- Rozproszenie położenia i stanu.



Procesy rozproszone



Trzy procesy
jednowątkowe

Jeden proces z trzema
wątkami

Procesy rozproszone

- Wyróżnia się dwa modele rozproszonych procesów:
 - Model statyczny
 - Model dynamiczny
- Koncepcja procesu
- Zarządzanie procesami
- Środowisko wykonywania procesów
- Przełączanie kontekstu procesów

Wędrówka procesów

- Daje możliwość przemieszczania procesów w trakcie ich wykonywania z jednego procesora na inny.
- Problemy:
 - Ustalenie stanu procesu: stan wewnętrzny, lokalna kolejka komunikatów, stan komunikacji ze zdalnymi procesami.
 - Transparentność wędrówki.
 - Obsługa przyszłej komunikacji.
 - Kiedy zdalnie wykonywać proces.

Modele wędrówki procesów

- Modele wędrówki procesów ze względu na przesyłanie informacji:
 - Przenośność słaba – segment kodu
 - Przenośność silna – segment kodu + segment wykonania
- Miejsce rozpoczęcia wędrówki:
 - Przenośność inicjowana przez nadawcę
 - Przenośność inicjowana przez odbiorcę

Wiązanie zasobów lokalnych

- Sposoby powiązania procesu z zasobem:
 - Wiązanie przez identyfikator
 - Wiązanie przez wartość
 - Wiązanie przez typ

Komunikaty a wędrówka procesów

- Sposoby postępowania:
 - Przekierowywanie komunikatów
 - Zapobieganie utracie komunikatów
 - Odzyskiwanie utraconych komunikatów