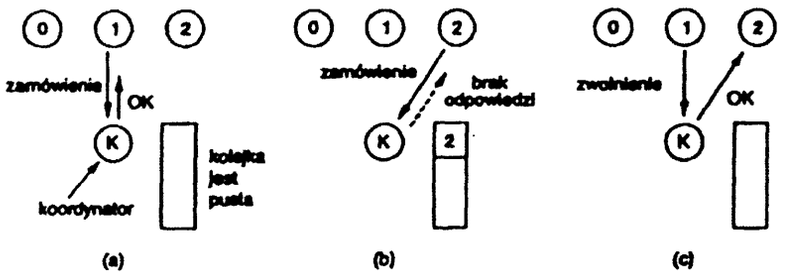
WZAJEMNE WYŁĄCZANIE W SYSTEMIE ROZPROSZONYM

Metody wykorzystujące pamięć dzieloną (semafory, monitory) nie są odpowiednie w systemach rozproszonych.

Algorytm scentralizowany

Jeden proces jest koordynatorem.
Proces, który chce wejść do sekcji krytycznej wysyła zamówienie do koordynatora.
Koordynator odpowiada (udziela zezwolenia), gdy żaden inny proces nie jest w sekcji krytycznej.
Proces, po odebraniu zezwolenia, wchodzi do sekcji krytycznej.
Proces wychodząc z sekcji krytycznej wysyła komunikat do koordynatora.



Cechy algorytmu scentralizowanego:
zapewnia wzajemne wyłączanie,
nie zachodzi głodzenie procesów,
łatwy w realizacji,
wrażliwy na awarie.

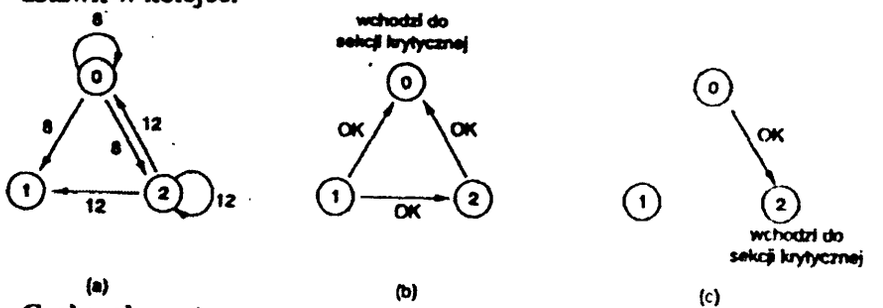
Algorytm rozproszony

Wymagane całkowite uporządkowanie czasowe zdarzeń - komunikatów (np. stosując algorytm Lamporta).

Proces, który chce wejść do sekcji krytycznej wysyła do wszystkich procesów komunikat zawierający nazwę sekcji krytycznej, swój numer, bieżący czas.
Każdy komunikat jest potwierdzany (zapewnienie niezawodności).
Proces odbierający komunikat:

1. Jeśli nie jest w sekcji krytycznej i nie chce do niej wejść - wysyła do nadawcy komunikat OK.
2. Jeśli jest w sekcji krytycznej - nie odpowiada.
3. Jeśli chce wejść do sekcji krytycznej - sprawdza znacznik czasu odebranego komunikatu i komunikatu, który sam wysłał. Jeśli odebrany komunikat ma znacznik czasu mniejszy - wysyła OK.

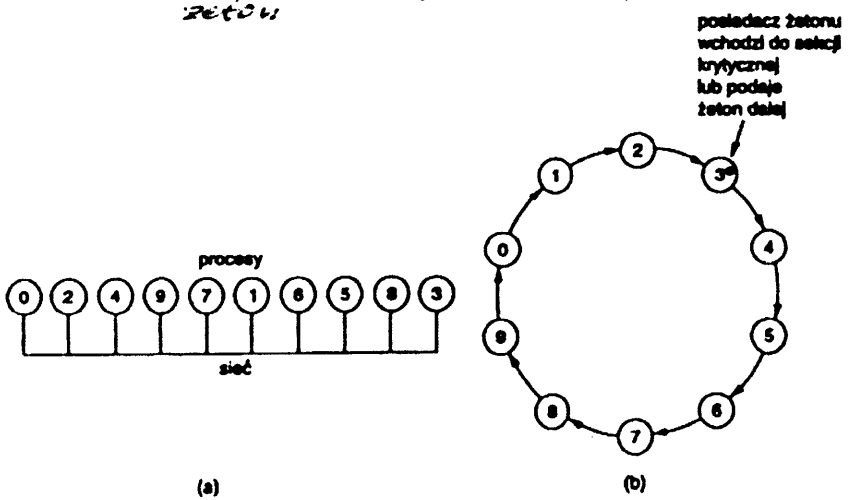
Proces nadawca:
Czeka aż wszystkie procesy udzielą zezwolenia, wtedy wchodzi do sekcji krytycznej.
Wychodząc z sekcji krytycznej wysyła OK. do procesów, które ustawił w kolejce.



Cechy algorytmu:
Zapewnienie wzajemnego wyłączania bez głodzenia.
Wrażliwość na awarie - brak odpowiedzi spowodowany awarią procesu jest traktowany jako brak zgody - blokowanie procesów próbujących wejść do sekcji krytycznej (jest możliwość rozwiązania).
Wymagana komunikacja grupowa lub każdy proces musi utrzymywać listę procesów znajdujących się w grupie, wchodzących i wychodzących - obciążenie systemu.

Algorytm pierścienia logicznego z żetonem

Rozpatrzmy system rozproszony, w którym zbiór procesów jest połączonych szyną.
Wprowadza się logiczne (programowe) uporządkowanie procesów tworząc pierścień.
W pierścieniu krąży żeton.
Proces po otrzymaniu żetonu sprawdza, czy chce wejść do sekcji krytycznej,
nie - przekazuje żeton sąsiadowi,
tak - zatrzymuje ~~pierścień~~ żeton, aż wyjdzie z sekcji krytycznej.



(a) (b)

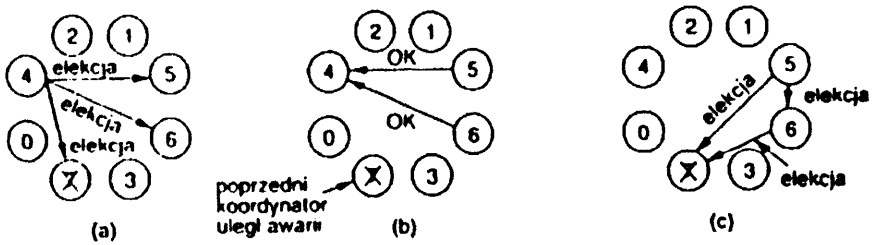
Cechy algorytmu:
Zapewnia wzajemne wyłączanie bez głodzenia.
Nie powoduje głodzenia procesów.
Problemy związane z zaginięciem żetonu.
Wrażliwość na awarie procesów.

ALGORYTMY ELEKCJI

Cel: wybór procesu, który będzie pełnił rolę koordynatora lub inicjatora w systemie rozproszonym.

Założenia:
każdy proces ma niepowtarzalny numer,
każdy proces zna numery wszystkich pozostałych,
procesy nie wiedzą, które z nich aktualnie działają, a które są unieruchomione,
próbuje się zlokalizować proces o największym numerze.

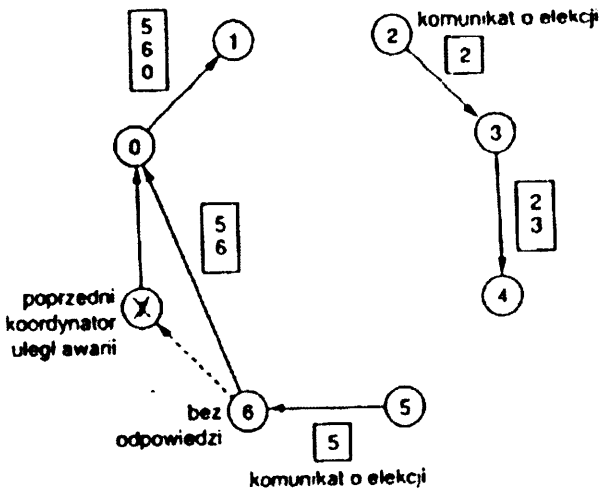
Algorytm tyrana:
1. Proces A zauważył, że koordynator nie odpowiada.
   A wysyła komunikat ELEKCJA do wszystkich procesów z **większymi** numerami.
2. Brak odpowiedzi, to A zostaje koordynatorem.
3. Nadchodzi komunikat od procesu B o większym numerze. Proces A przestaje działać w elekcji.
   B przejmuje sterowanie i kontynuuje elekcję (zgodnie z punktami 1, 2, 3).
4. Proces, który wygrywa elekcję wysyła do pozostałych komunikat: KOORDYNATOR.



Algorytm pierścieniowy

Założenie: procesy są fizycznie i logicznie uporządkowane. (Każdy proces przechowuje strukturę pierścienia).

Działanie:
1. Proces A zauważył, że koordynator nie działa. Wysyła komunikat ELEKCJA do kolejnego nie wyłączonego procesu w pierścieniu. Komunikat zawiera jego numer.
2. Proces B otrzymujący komunikat ELEKCJA, dopisuje swój numer i przesyła do następnego, itd.
3. Proces A po odebraniu komunikatu z własnym numerem, wysyła komunikat: KOORDYNATOR z pełną listą procesów występujących aktualnie w pierścieniu i wskazujący proces o najwyższym numerze będący koordynatorem.
4. Koordynator rozpoczyna działanie.



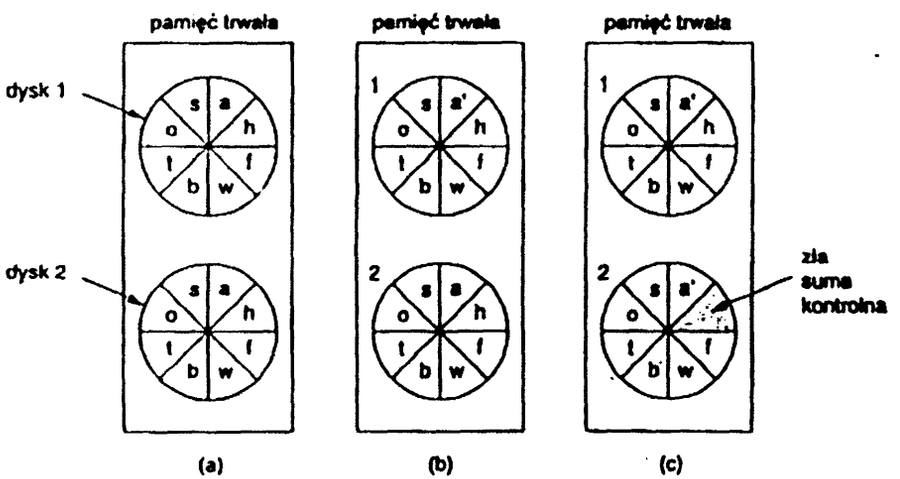
TRANSAKCJE NIEPODZIELNE
(atomic transactions)

Wprowadzenie
Ilustracja na przykładzie negocjacji i podpisania umowy handlowej.
Idea transakcji w systemie komputerowym
Zasada wszystko albo nic.

MODEL TRANSAKCJI

Założenia dotyczące systemu:
istnieje pewna liczba niezależnych procesów
każdy proces może ulec awarii
komunikacja między procesami jest zawodna

Rodzaje pamięci
RAM
Pamięci dyskowe
Pamięci trwałe (stable storage) specjalnie zaprojektowane, aby mogły przetrwać wszystkie awarie (z wyjątkiem kataklizmów). Są odpowiednie dla transakcji niepodzielnych.
Przykład realizacji pamięci trwałej na dwóch dyskach - rys.



(a) Pamięć trwała. (b) Awaria po zaktualizowaniu dysku 1. (c) Błędny obszar

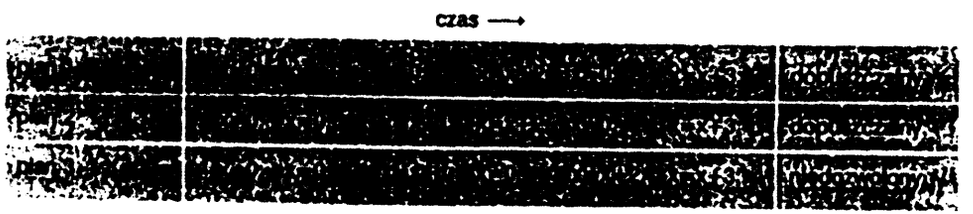
Transakcje elementarne
Elementarne działania niezbędne do programowania za pomocą transakcji, np.:
- Początek transakcji
- Koniec transakcji
- Zaniechanie transakcji
- Czytaj (dane z obiektu)
- Zapisz (dane do obiektu)

Właściwości transakcji
- **Niepodzielność** (atomicity)
  Transakcja albo jest realizowana w całości, albo wcale. Jeśli następuje, to tylko jako jedno niepodzielne i natychmiastowe działanie.
- **Spójność** (consistence)
  Nie są naruszane niezmienniki systemowe.
- **Izolacja, uszeregowanie** (isolation, serialization)
  Kilka transakcji może przebiegać w tym samym czasie. Wynik końcowy powinien być taki, jakby transakcje były wykonywane po kolei (etapowo).

```
POCZĄTEK_TRANSAKCJI
    x = 0;
    x = x + 1;
KONIEC_TRANSAKCJI
```
(a)

```
POCZĄTEK_TRANSAKCJI
    x = 0;
    x = x + 2;
KONIEC_TRANSAKCJI
```
(b)

```
POCZĄTEK_TRANSAKCJI
    x = 0;
    x = x + 3;
KONIEC_TRANSAKCJI
```
(c)



(d)

(a)-(c) Trzy transakcje. (d) Możliwe plany

• **Trwałość (durability)**
Po zatwierdzeniu transakcji jej wynik jest nieodwracalny i trwały.

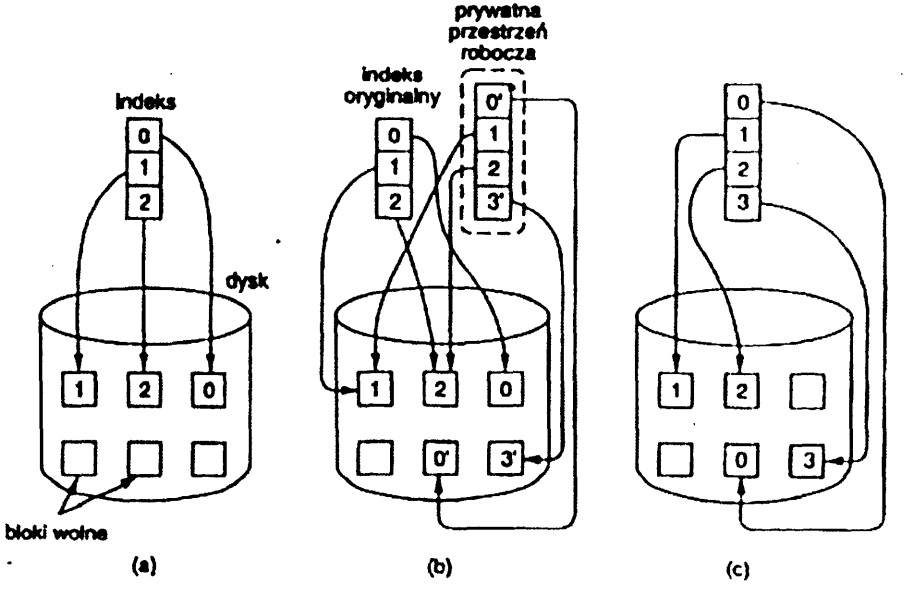
Zagnieżdżanie transakcji
Transakcja może się rozwidlać. Tworzone są wtedy transakcje pochodne, wykonywane równolegle na różnych maszynach.
Transakcje te również mogą się rozwidlać.
Zasada trwałości dotyczy tylko transakcji pierwotnej.
Może być wiele transakcji pochodnych, zagnieżdżonych dowolnie głęboko. Każda rozpoczęta transakcja dostaje prywatną kopię obiektów systemu i działa na kopiach (w swoim prywatnym świecie).

METODY REALIZACJI TRANSAKCJI

PRYWATNA PRZESTRZEŃ ROBOCZA

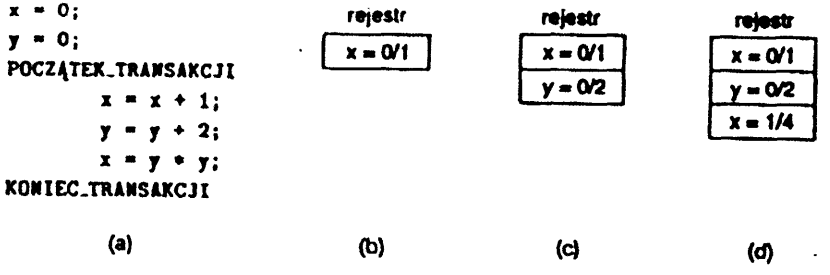
Proces rozpoczynający transakcję dostaje przydzieloną prywatną przestrzeń roboczą zawierającą kopie rzeczywistych obiektów.
W przypadku zatwierdzenia transakcji - zmiany przenoszone są obiekty rzeczywiste. Zaniechanie transakcji - usunięcie prywatnej przestrzeni roboczej.
Problem - wysoki koszt kopiowania wszystkich obiektów.
Rozwiązania: rozróżnienie obiektów do czytania i do aktualizacji, zastosowanie indeksowania.

Wkorzystanie indeksowania
Indeks (i-węzeł w syst. UNIX) zawiera adresy dyskowe bloków pliku.
Do prywatnej przestrzeni kopiuje się tylko indeks.
Czytanie pliku - odwołanie do oryginalnego pliku.
Aktualizacja bloku - stworzenie kopii bloku, wstawienie adresu do prywatnego indeksu, aktualizacja bloku.
Dodanie bloku (shadow block) - dostawienie adresu nowego bloku do prywatnego indeksu.
Zatwierdzenie transakcji - przemieszczenie prywatnego indeksu do przestrzeni procesu rodzicielskiego.



REJESTR ZAPISÓW WYPRZEDZAJĄCYCH - LISTA ZAMIARÓW (writeahead log, intentions lists)

Idea:
Pliki są modyfikowane w miejscu ich występowania, ale przed zmianą jakiegokolwiek bloku następuje zapisanie rekordu w specjalnym rejestrze zapisów wyprzedzających, przechowywanym w pamięci trwałej.
Zapisy obejmują: transakcję, która dokonuje zmiany, jaki plik i blok jest zmieniany, starą i nową zawartość bloku.



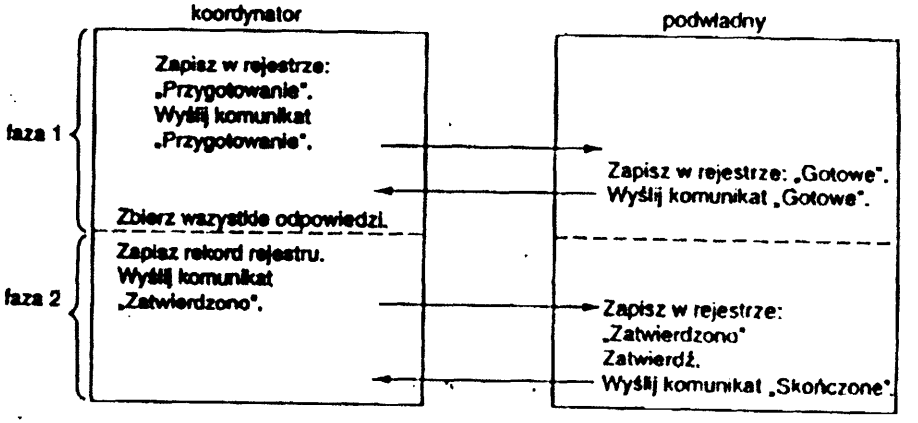
Rys. Ilustracja zapisów w rejestrze transakcji używającej dwóch obiektów x, y.

Postępowanie w różnych sytuacjach:

Zatwierdzenie transakcji - do rejestru wpisywany jest rekord zatwierdzenia. Zmiany w plikach są już dokonane.
Zaniechanie transakcji - wycofanie (rollback), t.j. przywrócenie stanu początkowego na podstawie zapisów w rejestrze.
Awaria - rejestr umożliwia rekonstrukcję danych, możliwe jest kontynuowanie transakcji lub jej odwołanie

PROTOKÓŁ ZATWIERDZANIA DWUFAZOWEGO (two-phase commit protocol)

Rozwiązanie problemu zapewnienia niepodzielności transakcji w sytuacji współpracy wielu procesów na różnych maszynach. Każdy z procesów może przechowywać część obiektów modyfikowanych przez transakcję.
Idea:
Jeden z procesów jest koordynatorem , pozostałe - podwładnymi.
Zastosowanie specjalnego protokołu zatwierdzania, wykorzystującego wymianę komunikatów między procesem koordynatorem, a procesami podwładnymi.
Potwierdzanie wszystkich działań zapisami w rejestrze.



Rys. Ilustracja wymiany komunikatów dwufazowego zatwierdzania transakcji

Uwagi:
Rejestr przechowywany jest w pamięci trwałej.
Zapis aktualnego stanu w rejestrze umożliwia kontynuację działań także w przypadku awarii - odporność na awarie.

ALGORYTMY NADZOROWANIA WSPÓŁBIEŻNOŚCI
(concurrency control algorithms)

Mechanizmy nadzorujące jednoczesne wykonywanie transakcji przez wiele procesów na różnych maszynach, w przypadku przetwarzania tych samych obiektów

Takie nadzorowanie, aby nie wystąpiły zakłócenia

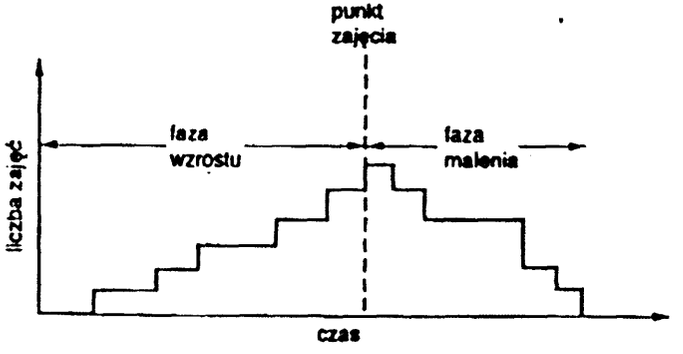
Zajmowanie (locking)

Proces wykonujący transakcję zamyka obiekt, np. plik przed korzystaniem z niego przez inne procesy. Operacja zajmowania zarządzana jest przez pewien proces lub procesy zarządzające. Zarządca aktualizuje listę zajętych obiektów. Nie pozwala innym procesom na dostęp do obiektów zajętych.

Możliwość rozróżnienia zajmowania obiektu do czytania i do aktualizacji.

Ziarnistość zajmowania (granularity) - wielkość zajmowanego obiektu. Obiektem może być jednostka znacznie mniejsza niż plik np. rekord lub strona, lub większa np. cała baza danych.

Zajmowanie dwufazowe (two-phase locking) schemat zajmowania obiektów, wg. którego w pierwszej fazie proces tylko zajmuje wszystkie niezbędne obiekty (faza wzrostu), a w drugiej fazie (fazie malenia) - tylko zwalnia.



Uwagi:

Udowodniono, że jeśli do wszystkich transakcji stosuje się zajmowanie dwufazowe, to plany realizacji tych transakcji są szeregowalne. Mogą wystąpić blokady.

Optymistyczne nadzorowanie współbieżności
(optimistic concurrency control)

Metoda stosowana zwłaszcza przy wykorzystywaniu prywatnych przestrzeni roboczych.

Polega na zapisywaniu informacji, które obiekty były aktualizowane. Wykonuje się transakcję nie zważając na inne. W chwili zatwierdzania transakcji sprawdza się czy inna transakcja nie zmodyfikowała plików po jej rozpoczęciu, jeśli tak - zaniechanie transakcji, jeśli nie - zatwierdzenie.

Ma wiele zalet w sytuacjach, gdy rzadko występują konflikty.

Odporna na blokady, żaden proces nie musi czekać na zajęte obiekty.

Wada - konieczność powtórzenia całej transakcji w przypadku modyfikacji obiektów po jej rozpoczęciu przez inną transakcję

Ziarnistość zajmowania - jak duży jest obiekt który ma być zajęty.

Algorytmy te mają zapewnić własności izolacji i uszeregowania - zajmowanie dwufazowe.

Jeśli proces zajął jakiś obiekt i mógłby go zwolnić, to może to zrobić dopiero wtedy, gdy wszystkie obiekty zostaną zajęte. Gdy to jest zapewnione to mamy izolację, uszeregowanie procesów

Zaniechanie transakcji - powrót do początku

Rozpatrzyć plany współbieżnej realizacji transakcji

a) początek transakcji

$x=0,\ x=x+11,\ y=1,\ y=x+y;$

koniec transakcji

b) początek transakcji

$x=0,\ x=x+5,\ y=3$

koniec transakcji

c) początek transakcji

$x=0,\ y=0,\ x=x+7,\ y=y+2-x$

koniec transakcji

początek transakcji

pobranie (kwota, konto 1)

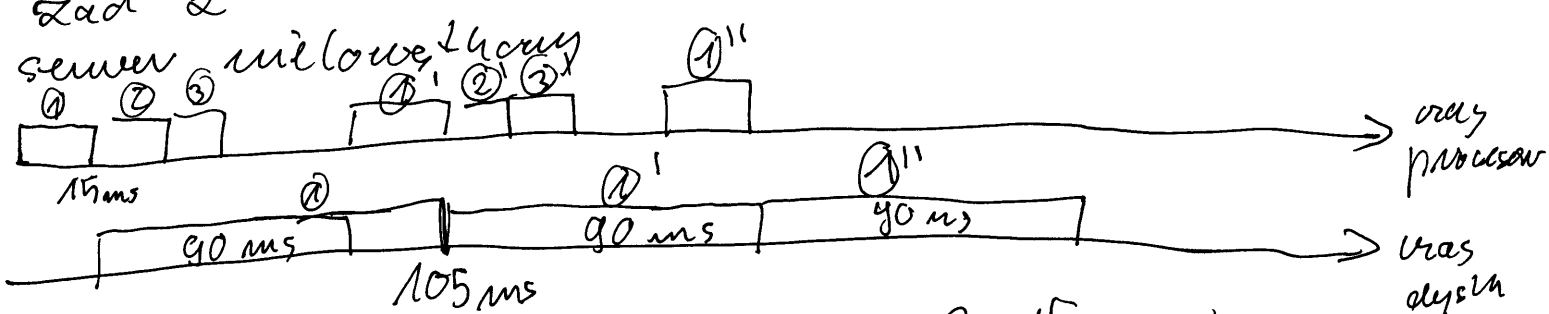
wypłata (kwota, konto 1);

koniec transakcji

1. Proszę określić pełną listę dopuszczalnych wartości zmiennych x i y (rozdzielając je słowami "i" "lub".

2. Podać przykład niedopuszczalnego planu (ze względu na własności izolacji)

Zad 2



$1000\ ms = \frac{1}{3} z (15\ ms + 90\ ms - 3 \cdot 15\ ms) +$

$+ \frac{2}{3} z \cdot 15\ ms$

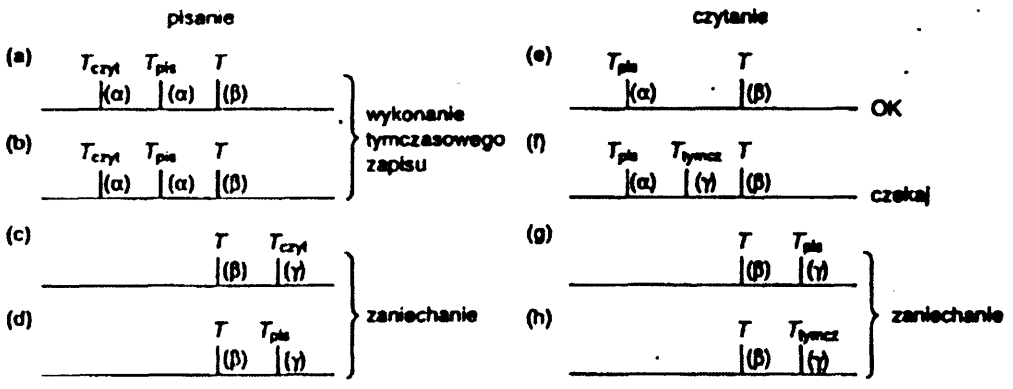
$1000\ ms = 30\ ms \cdot z$

$z = 33{,}3$ zam/sek

Wykorzystanie procesora 50%

Znaczniki czasu

Każdej transakcji przypisany jest znacznik czasu operacji elementarnej „Początek transakcji".
Zapewniona jest niepowtarzalność znaczników czasu (algorytm Lamporta).
Każdy plik ma skojarzony znacznik czasu czytania i znacznik czasu pisania przez ostatnią zatwierdzoną ttransakcję.
Znacznik czasu czytania i pisania do pliku mniejsze od znacznika czasu danej transakcji - nie ma problemu.
Sytuacja odwrotna oznacza, że po rozpoczęciu transakcji, inna, późniejsza transakcja miała dostęp do pliku. Sposób postępowania wyjaśnia rysunek.



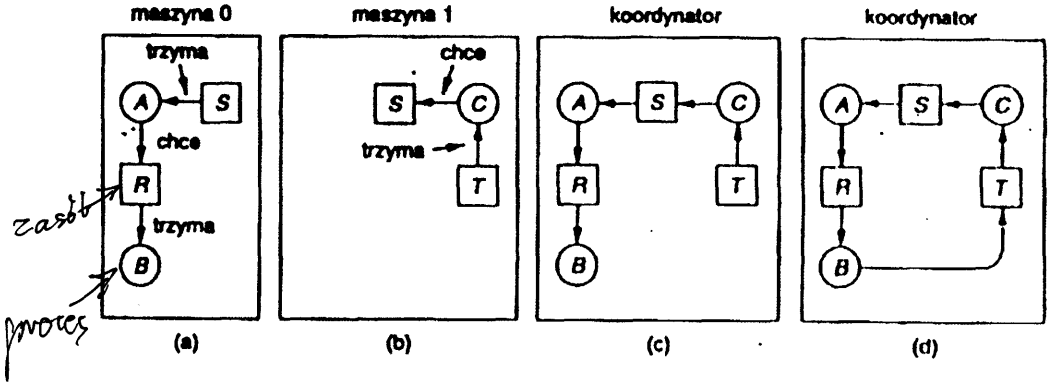
Uwagi:
Stosowanie znaczników powoduje odmienne postępowanie niż w przypadku metody zajmowania. Transakcja, która spotka późniejszy znacznik musi być zaniechana.
Stosowanie znaczników jest bezpieczniejsze - zapobiega powstawaniu blokad.

BLOKADY W SYSTEMACH ROZPROSZONYCH

Metody postępowania - analogiczne jak w przypadku systemów jednoprocesorowych:
- Zapobieganie.
- Unikanie.
- Wykrywanie i usuwanie skutków.
- Ignorowanie problemu.

Przykłady postępowania

Scentralizowane wykrywanie blokady
System: zbiór maszyn, jeden koordynator.
Każda maszyna utrzymuje graf własnych procesów i zasobów.
Koordynator tworzy graf dla całego systemu.
Sposoby przesyłania informacji:
1. Maszyna wysyła komunikat po każdej zmianie krawędzi w grafie.
2. Maszyna wysyła okresowo wykaz dodanych i usuniętych krawędzi.
3. Koordynator prosi maszyny o przesłanie informacji, gdy będzie mu to potrzebne.

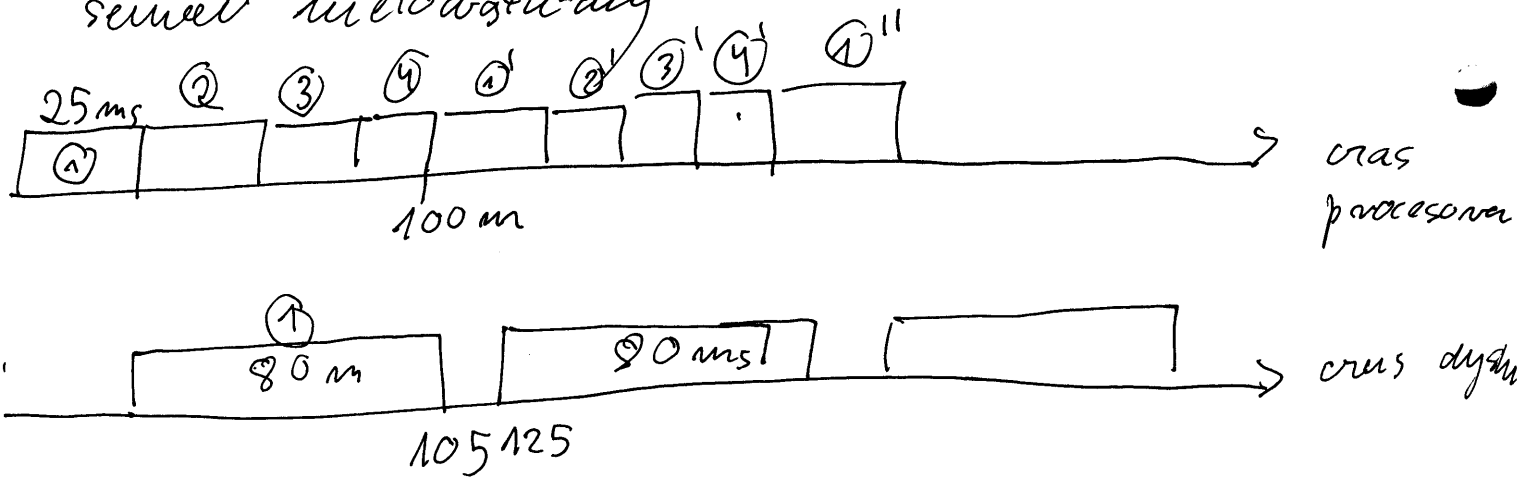


Rys. ilustruje możliwość powstania tzw. **fałszywej blokady.**
Poprawne rozwiązanie:
Logiczna synchronizacja czasu (algorytm Lamporta)
Komunikaty zawierają znaczniki czasu.
Koordynator może wyeliminować tworzenie fałszywych blokad.

Fałszywa blokada, jeśli któryś z komunikatów o zwolnieniu jakiegoś zasobu nie dojdzie, wówczas koordynator tworzy cykl w grafie i który tworzy blokadę

Zadanie 1

serwer wielowątkowy



$$1000\,ms = \frac{1}{4} z\,(25\,ms + 80\,ms - 80\,ms) +$$

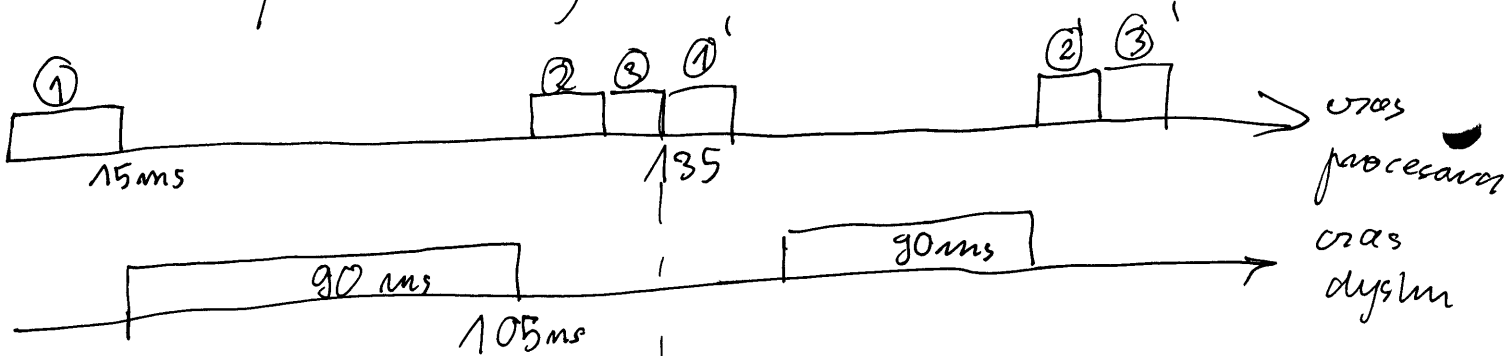
$$= \frac{3}{4} z \cdot 25\,ms$$

$$1000\,ms = 25\,ms \cdot z$$

$$z = 40 \text{ zamówień / 1 sekundę}$$

Zadanie 2

serwer jednowątkowy



z – liczba zamówień obsługiwanych w ciągu 1 sek

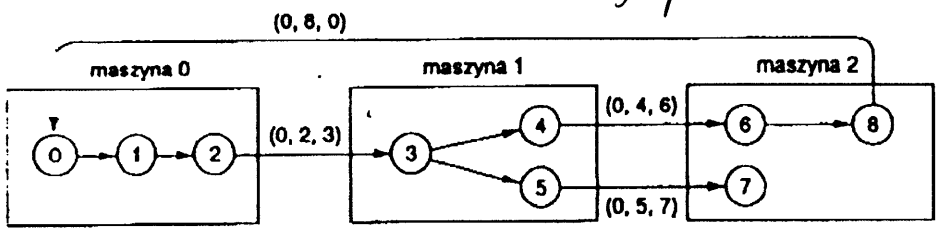
$$1000\,ms = \frac{1}{3} z\,(15\,ms + 90\,ms) + \frac{2}{3} z\,(15\,ms)$$

$$1000\,ms = 45\,ms \cdot z$$

$$z = \frac{1000\,ms}{45} = 22 \text{ zamówień}$$

Rozproszone wykrywanie blokady

Przykład algorytmu Chandy-Misra- Haasa.
Proces może zamawiać wiele zasobów jednocześnie.
Cały czas asynchroniczny przesyłanie komunikatów
Proces 0 oczekuje na zasób, który przetrzymuje proces 1



Na rysunku tylko grafy oczekiwań.
Idea:
Proces oczekujący na zasób wysyła komunikat do procesu przetrzymującego ten zasób.
Komunikat zawiera:
Nr procesu rozpoczynającego czekanie.
Nr procesu wysyłającego komunikat.
Nr procesu, do którego komunikat jest wysyłany.

Odbiorca komunikatu sprawdza czy sam nie czeka, jeśli czeka to wysyła kolejny komunikat aktualizując 2-ie i 3-e pole.

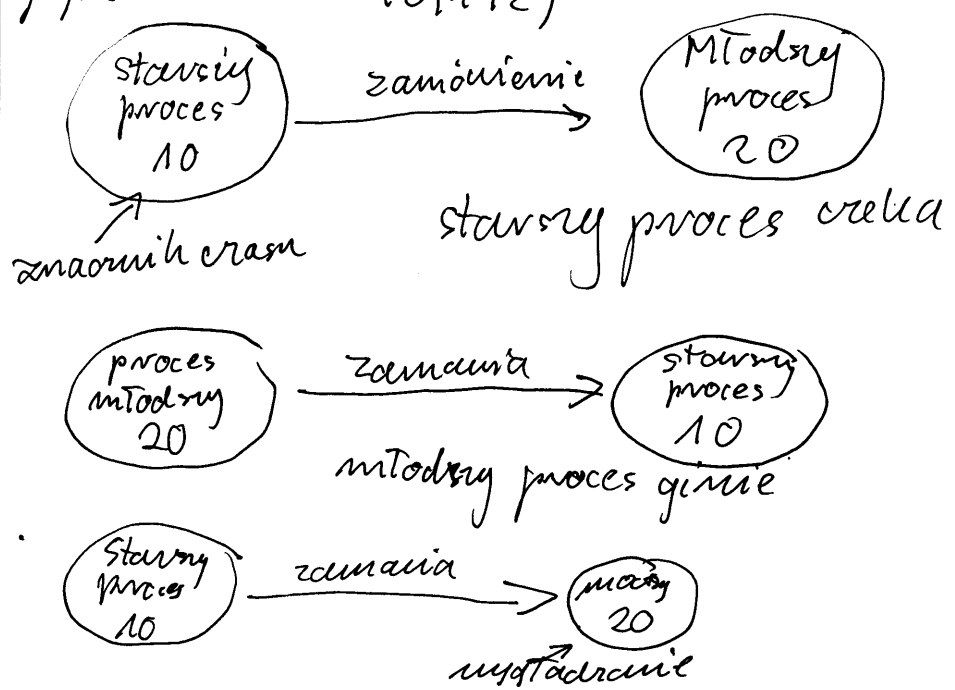
Powrót komunikatu do pierwotnego nadawcy oznacza blokadę. (cykl)

Sposób usunięcia blokady: np. usunięcie procesu, który zapoczątkował próbę.

Zapobieganie blokadom w systemach rozproszonych - przykład praktycznego podejścia wykorzystującego niepodzielne transakcje i znaczniki czasu

Każda transakcja ma jednoznacznie przyporządkowany niepowtarzalny znacznik czasu. (algorytm Lamporta)
Idea:
Jeśli proces zamawia zasób przetrzymywany przez inny proces - sprawdza się znaczniki czasu.
Sposób postępowania:
Proces starszy zamawia zasób przetrzymywany przez proces młodszy - proces starszy czeka.
Proces młodszy zamawia zasób przetrzymywany przez proces starszy - proces młodszy ginie.
Usunięcie procesu starszego jest bardziej kosztowne
Własność algorytmu: znaczniki procesów oczekujących w łańcuchu są rosnące, zapobiega to powstaniu cyklu.

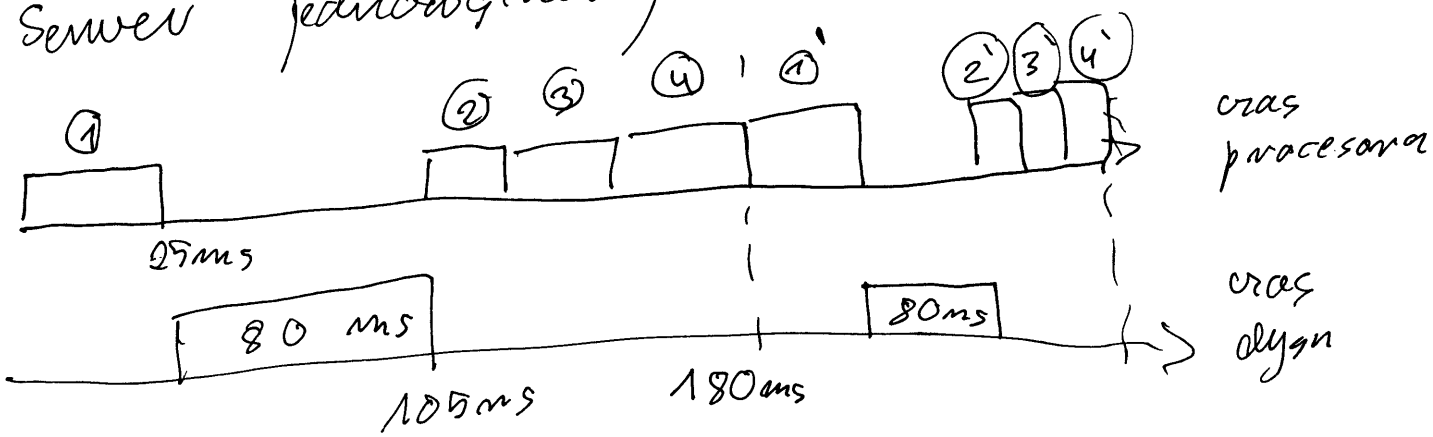
Proces 0 wysyła komunikat (0,0,1)
Proces 1 wysyła komunikat (0,1,2)



Zadanie 1 a

Najpierw przydzielę zamówienia z operacją dyskową

Serwer jednowątkowy



z - liczba zamówień obsłużonych w czasie 1s

$1\ s = 1000\ ms =$

$= \frac{1}{4} z \cdot (25\ ms + 80\ ms) + \frac{3}{4} z \cdot 25\ ms = 45\ ms \cdot z =$

$= \frac{1000}{45} \approx 22$ zamówień / 1 sekundę

$1000\ ms = 45\ ms \cdot z$

procent wykorzystania procesora

$$\frac{4{,}25\ ms}{180} \cdot 100\%$$

~~zad~~ 2

PROCESY I WĄTKI W SYSTEMACH ROZPROSZONYCH

Proces z jednym wątkiem sterowania:
własny licznik rozkazów, stos, zbiór rejestrów, przestrzeń adresowa;
komunikacja między procesami - systemowe mechanizmy komunikacji (np. semafory, komunikaty).

Wiele wątków sterowania w procesie:
każdy wątek ma własny licznik rozkazów, stos, rejestry,
ale wszystkie wątki mają wspólną przestrzeń adresową, ten sam zbiór otwartych plików, procesów pochodnych itp.

Cechy analogiczne (wątków i tradycyjnych procesów jednowątkowych):
wykonywanie sekwencyjne,
działanie współbieżne - podział czasu procesora,
stany wątków: gotowy, wykonywany, czekający (blokowany), likwidowany;
tworzenie wątków pochodnych.

Tab. Elementy procesu, w którego skład wchodzi wiele wątków (wspólne dla wszystkich wątków) oraz elementy pojedynczego wątku.

Proces	**Wątek**
przestrzeń adresowa	licznik rozkazów
zmienne globalne	stos
otwarte pliki	zbiór rejestrów
procesy pochodne	wątki pochodne
czasomierze	stan
sygnały	
semafory	
informacje rozrachunkowe	

Wątek stworzeniu - kolejność zarządzania. według której procesy są pobierane ze stosu i wykonywane

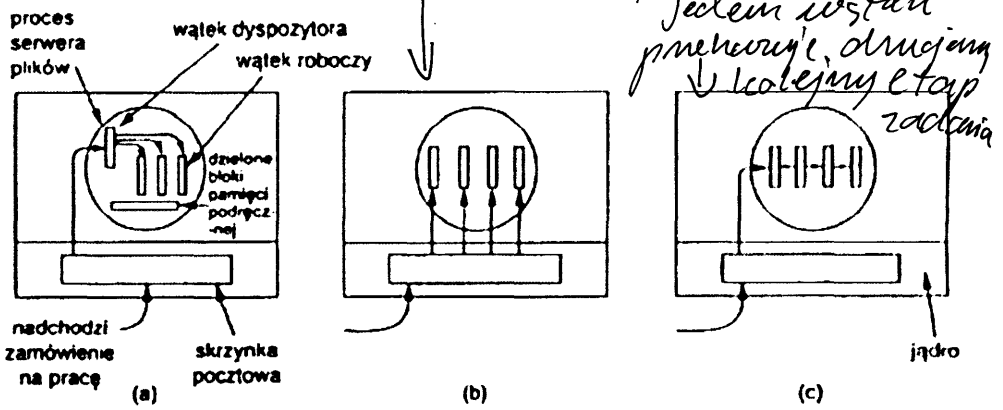
Sposoby organizacji wątków w procesie

Dispatcher - worker

Team model

Pipeline model

model zespołu, mamy wiele wątków i każdy co pewien czas sprawdza główną, jeśli znajdzie zadanie to je wykonuje i jedem wątek przekazuje drugiemu kolejny etap zadania



Trzy sposoby organizacji wątków w procesie: (a) dyspozytor-pracownik, (b) model zespołowy, (c) model potokowy

a) Serwer plików, od klientów przychodzą zamówienia do skrzynki pocztowej. Jest jedem wątek dyspozytora i wiele wątków roboczych. Jak znajdzie zamówienie, to budzi jeden z uśpionych wątków roboczych, który wtedy realizuje zamówienie. Każde zadanie może być obsługiwane przez inny wątek. Wątki działają współbieżnie.

Zadanie 1

Proszę porównać operację czytania pliku za pomocą jednowątkowego serwera plików i serwera wielowątkowego. Otrzymanie żądania dla pracy i skierowanie go do wykonania i reszta niezbędnego przetwarzania zajmuje 25 ms, pod warunkiem, że potrzebne dane znajdują się w podręcznej pamięci bloków. Jeśli istnieje konieczność wykonania operacji dyskowej, co stanowi jedną czwartą żądań, potrzeba dodatkowo 80 ms podczas gdy wątek jest uśpiony

Ile żądań na sekundę może obsłużyć serwer jednowątkowy?

Ile żądań na sekundę może obsłużyć serwer wielowątkowy?

Zadanie 2

Proszę porównać operację czytania pliku za pomocą jednowątkowego serwera plików i serwera wielowątkowego. Otrzymanie żądania na pracę i skierowanie zamówienia do wykonania i reszta niezbędnego przetwarzania zajmuje 15 ms, pod warunkiem, że potrzebne dane znajdują się w podręcznej pamięci bloków. Jeżeli istnieje konieczność wykonania operacji dyskowej, co stanowi jedną trzecią żądań, potrzeba dodatkowo 90 ms, podczas których wątek jest uśpiony.

Ile żądań na sekundę może obsłużyć serwer jednowątkowy?

Ile żądań na sekundę może obsłużyć serwer wielowątkowy