

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

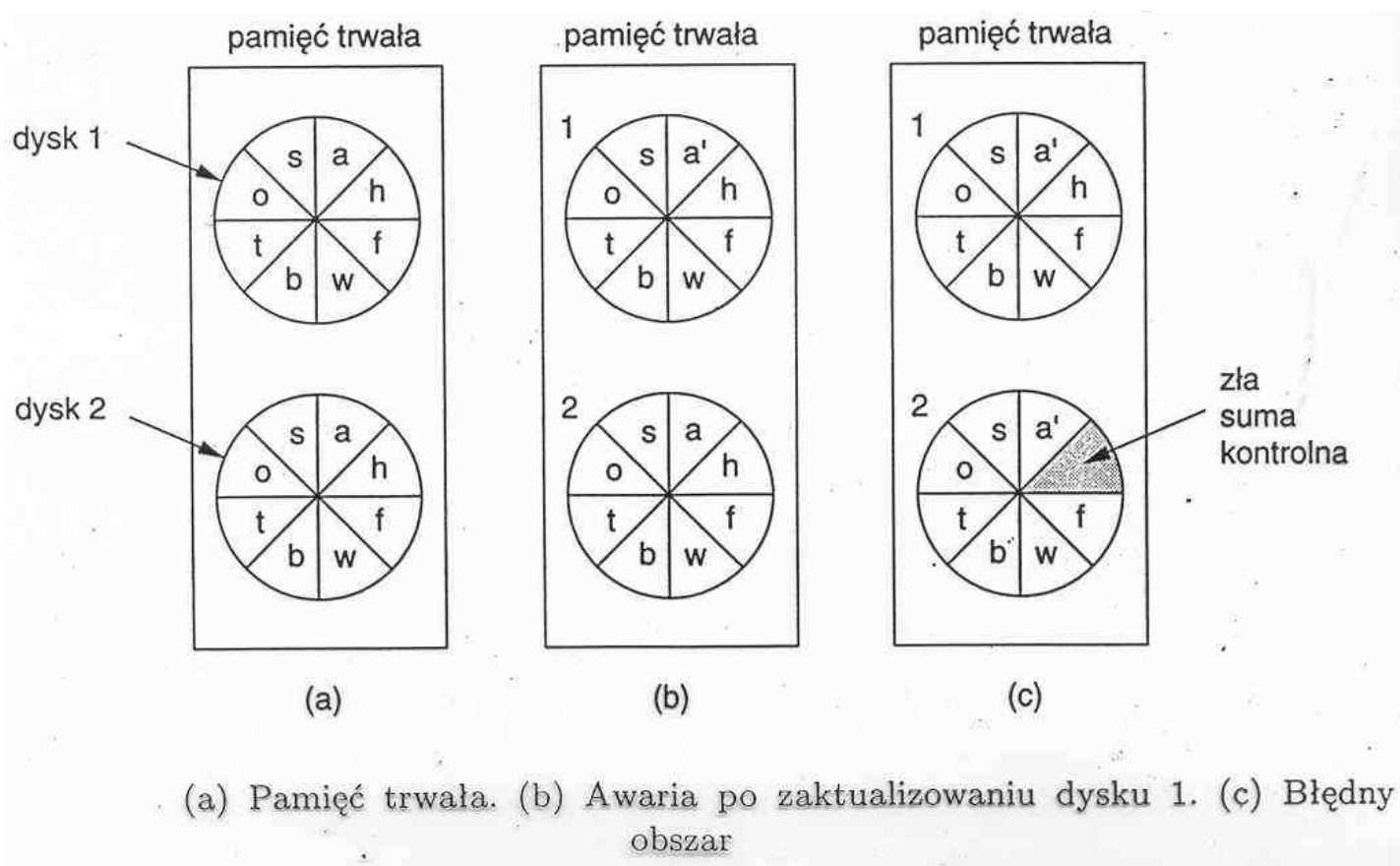
Rodzaje pamięci

RAM

Pamięci dyskowe

Pamięci trwałe (stable storage) specjalnie zaprojektowane, aby mogły przetrwać możliwie wszystkie awarie (z wyjątkiem kataklizmów). Potrzebne dla realizacji transakcji niepodzielnych.

Przykład realizacji pamięci trwałej na dwóch dyskach:



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

1. Niepodzielność (atomicity)

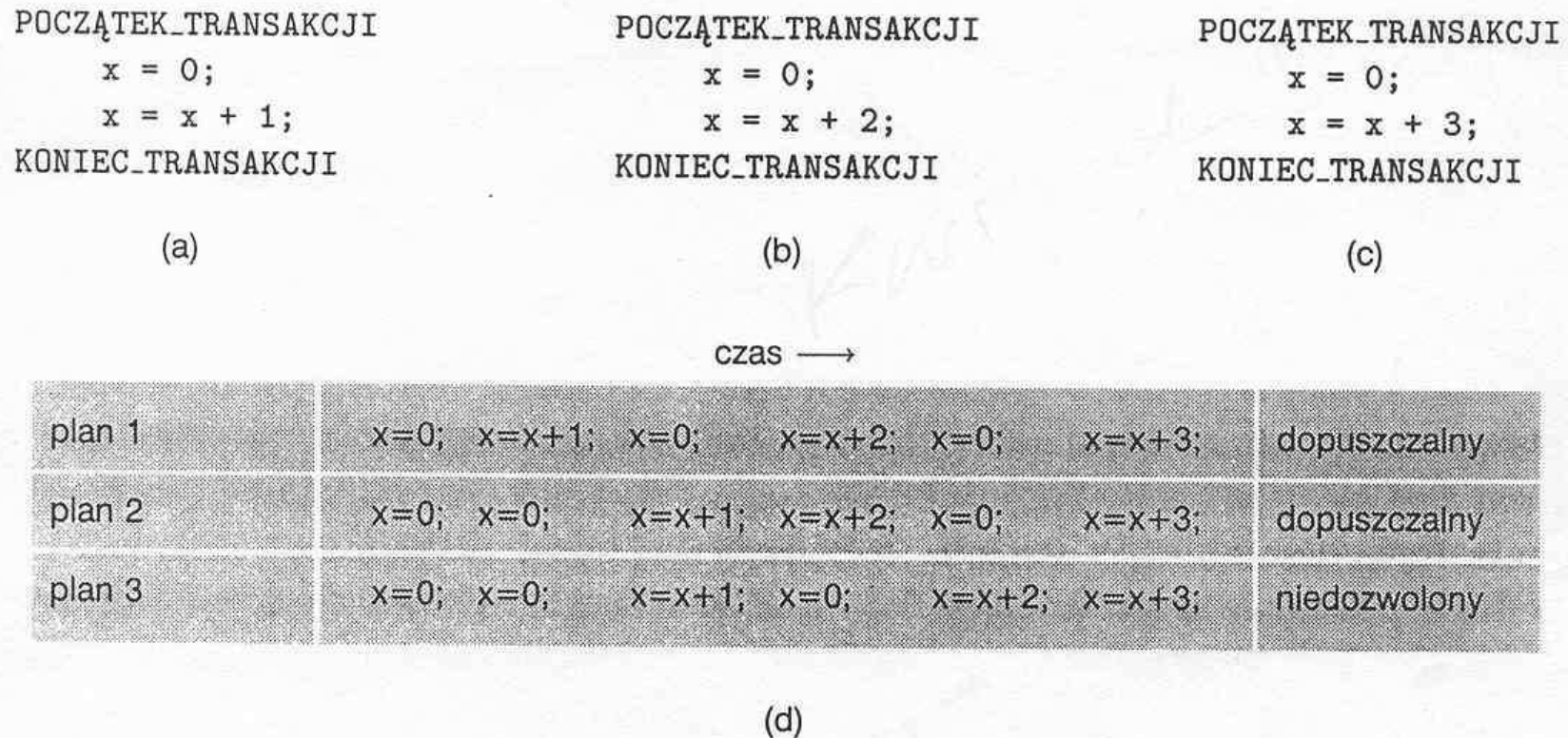
Transakcja albo jest realizowana w całości, albo wcale. Jeśli następuje, to tylko jako jedno niepodzielne i natychmiastowe działanie.

2. Spójność (consistence)

Nie są naruszane niezmienniki systemowe.

3. Izolacja, uszeregowanie (isolation, serialization)

Kilka transakcji może przebiegać w tym samym czasie. Wynik końcowy powinien być taki, jakby transakcje były wykonywane po kolei (etapowo).



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

4. Trwałość (durability)

Po zatwierdzeniu transakcji jej wynik jest nieodwracalny i trwały.

Uwagi dotyczące zagnieżdżania 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.

Jeden ze sposobów realizacji transakcji: 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.

Możliwe rozwiązanie: rozróżnienie obiektów do czytania i do aktualizacji, zastosowanie indeksowania.

Wykorzystanie indeksowania

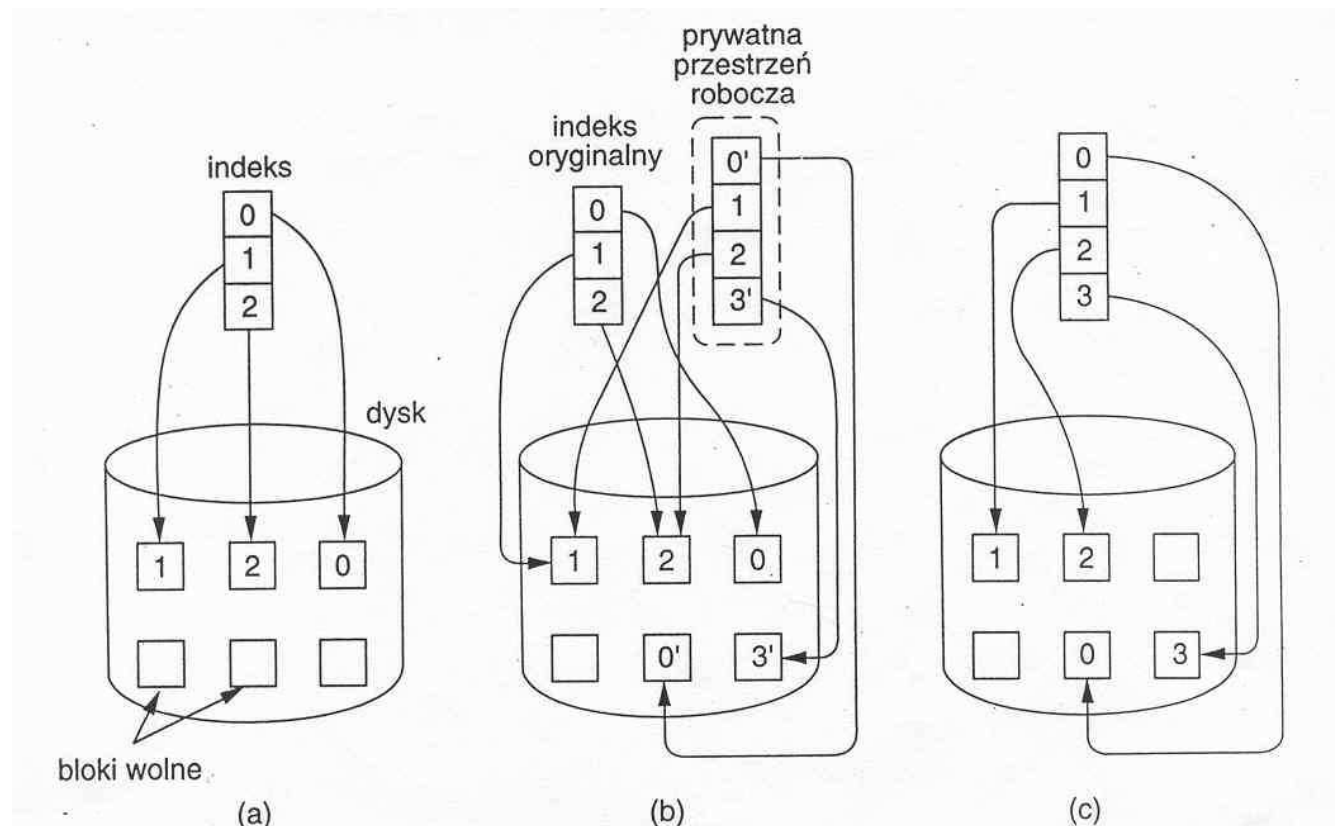
Indeks (i-węzeł w syst. UNIX) zawiera adresy dyskowe bloków pliku.

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.

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

x = 0; y=0;	rejestr	rejestr	rejestr
	x = 0/1	x =0/1	x=0/1
POCZĄTEK_TRANSAKCJI		y= 0/2	y = 0/2
x = x + 1;			x = 1/4
y = y + 2;			
x = y * y;			
KONIEC_TRANSAKCJI			

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 DWUFAZOWEGO ZATWIERDZANIA TRANSAKCJI **(two-phase commit protocol)**

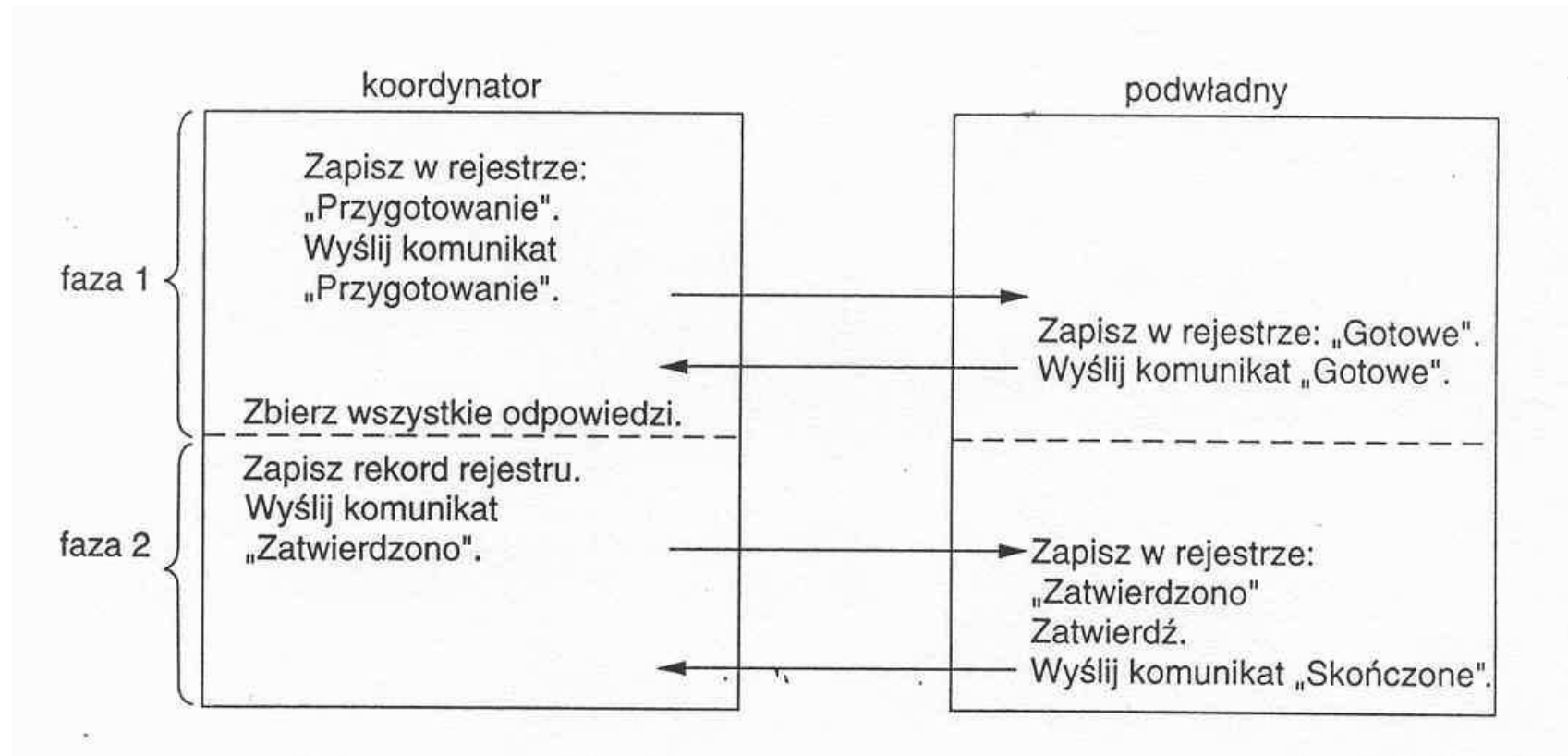
Rozwiązanie problemu zapewnienia niepodzielności transakcji w sytuacji współpracy wielu procesów na różnych maszynach.
Każdy z procesów może przechowywać część obiektów modyfikowanych przez transakcję.

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 przechowywanym w pamięci trwałej.



Ilustracja wymiany komunikatów dwufazowego zatwierdzania transakcji.

W fazie 1 - przygotowania - koordynator zleca podwładnym uczestnikom głosowanie za lub przeciw zatwierdzeniu transakcji.

W fazie 2 – podjęcie decyzji.

Uwagi:

Wszystkie decyzje zapisywane są w rejestrze.

Rejestr przechowywany jest w pamięci trwałej.

Zapis aktualnego stanu w rejestrze umożliwia kontynuację działań także w przypadku awarii.

Postępowania w przypadku awarii

Stany:

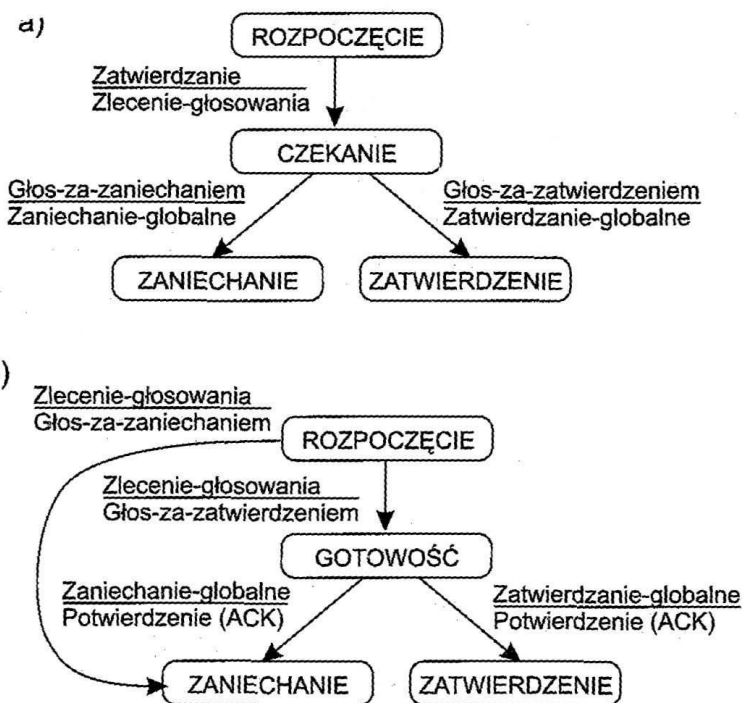
- a. maszyny koordynatora oraz
- b. maszyny uczestnika w protokole zatwierdzania dwufazowego

Możliwe sytuacje blokowania

- blokowanie się koordynatora czekającego na głos uczestnika – przekroczenie ustalonego czasu – prowadzi do decyzji o zaniechaniu,
- blokowanie się uczestnika, długo czekającego po rozpoczęciu na zlecenie głosowania – decyzja o zaniechaniu i odpowiedni komunikat do koordynatora,
- blokowanie uczestnika czekającego na komunikat z wynikiem głosowania – jeśli komunikat nie nadejdzie w określonym czasie – decyzja o zaniechaniu.

Przechowywanie wszystkich decyzji w pamięci trwałej umożliwia rekonstrukcję procesu z przesyłaniem dodatkowych komunikatów.

Inne rozwiązanie - protokół zatwierdzanie trzyczynowe (Skeen 1981).



ALGORYTMY NADZOROWANIA WSPÓŁBIEŻNOŚCI (concurrency control algorithms)

Mechanizmy nadzorujące współbieżne wykonywanie transakcji korzystających z tych samych danych, realizowanych przez wiele procesów na różnych maszynach.

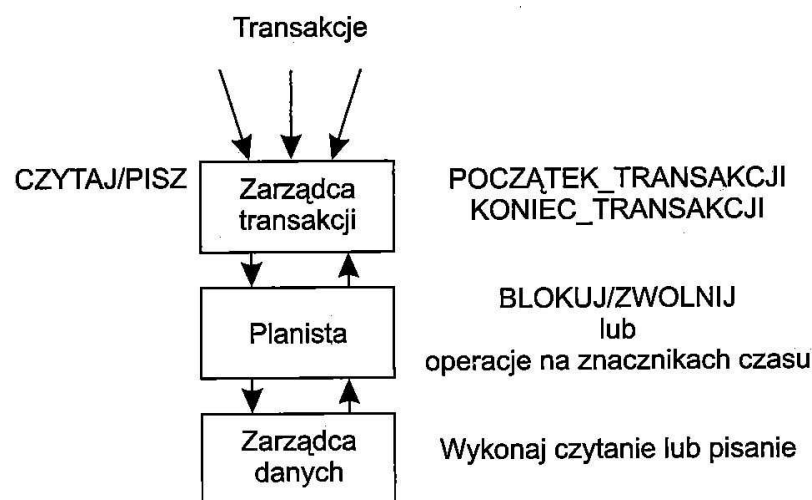
Zajmowanie – blokowanie (locking)

Proces wykonujący transakcję zamyka – blokuje obiekt, np. plik przed korzystaniem z niego przez inne procesy. Operacja blokowania zarządzana jest przez procesy zarządzające:

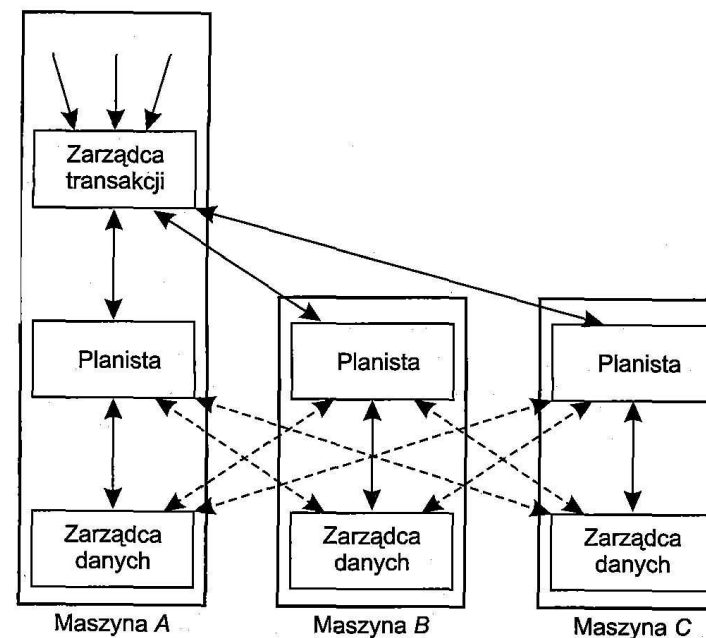
Zarządca transakcji - odpowiedzialny za niepodzielność transakcji.

Planista – odpowiada za właściwe sterowanie współbieżnością.

Zarządca danych – wykonuje rzeczywiste operacje czytania i zapisywania danych.



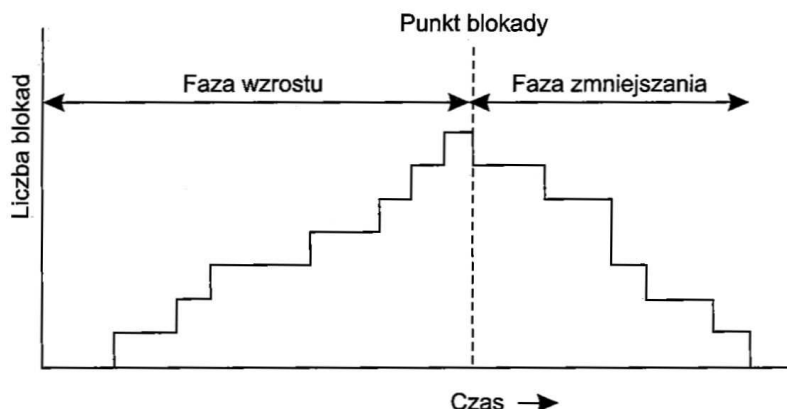
Ogólna organizacja zarządców



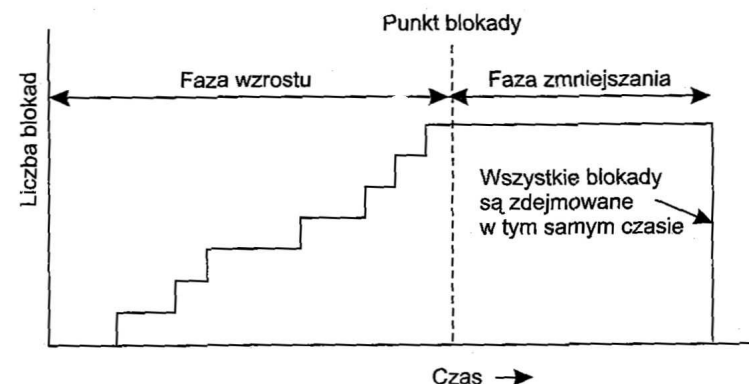
Ogólna organizacja zarządców obsługi
transakcji rozproszonych

Blokowanie dwufazowe (two-phase locking)

Schemat zajmowania obiektów, wg którego w pierwszej fazie proces tylko zajmuje – blokuje wszystkie niezbędne obiekty (faza wzrostu), a w drugiej fazie (fazie malenia) - tylko zwalnia.



Blokowanie dwufazowe



Ścisłe blokowanie dwufazowe

Uwagi:

Udowodniono, że jeśli do wszystkich transakcji stosuje się blokowanie dwufazowe, to plany realizacji tych transakcji są szeregowalne. Mogą jednak wystąpić zakleszczenia.

Zalety ścisłego blokowania dwufazowego: eliminacja „zaniechania kaskadowego” – konieczności anulowania zatwierdzonej transakcji z tego powodu, że miała kontakt z daną, której oglądać nie powinna, uproszczenie zarządzania.

Warianty rozwiązań w systemach rozproszonych :

- blokowanie scentralizowane - za nakładanie blokad odpowiada centralny zarządca,
- blokowanie z kopią podstawową – dla każdej danej jest określona kopia podstawowa, zarządca blokowania w maszynie z tą kopią odpowiada za nakładanie i zdejmowanie blokad,
- blokowanie rozproszone ze zwielokrotnieniem danych na wielu maszynach – planiści na każdej maszynie odpowiadają za blokowanie , a także za przekazywanie operacji lokalnemu zarządcy danych.

Optymistyczne nadzorowanie współbieżności ze znacznikami czasu (optimistic concurrency control)

Metoda stosowana zwłaszcza przy wykorzystywaniu prywatnych przestrzeni roboczych.

Polega na zapisywaniu informacji, które obiekty były czytane i zapisywane, ze znacznikami czasu operacji. Wymagana jest logiczna synchronizacja czasu w systemie.

Wykonuje się transakcję nie zważając na inne.

W chwili zatwierdzania transakcji sprawdza się, czy inna transakcja nie zmodyfikowała danych po jej rozpoczęciu (na podstawie znaczników czasu):

- jeśli tak – zaniechanie transakcji,
- jeśli nie - zatwierdzenie.

Zalety w sytuacjach, gdy rzadko występują konflikty.

- Odporna na zakleszczenia, żaden proces nie musi czekać na zajęte obiekty.

- Możliwość maksymalnego zrównoleglenia działań.

Wada: konieczność powtórzenia całej transakcji, gdy po jej rozpoczęciu, inna transakcja zmodyfikowała wykorzystywane obiekty danych.

Pesymistyczne nadzorowanie współbieżności z zastosowaniem znaczników czasu

Każdej transakcji przypisany jest znacznik czasu operacji elementarnej "Początek transakcji".

Zapewniona jest niepowtarzalność znaczników czasu (logiczna synchronizacja czasu - algorytm Lamporta).

Każdy plik ma skojarzony znacznik czasu czytania i znacznik czasu pisania przez ostatnią zatwierdzoną transakcję.

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.

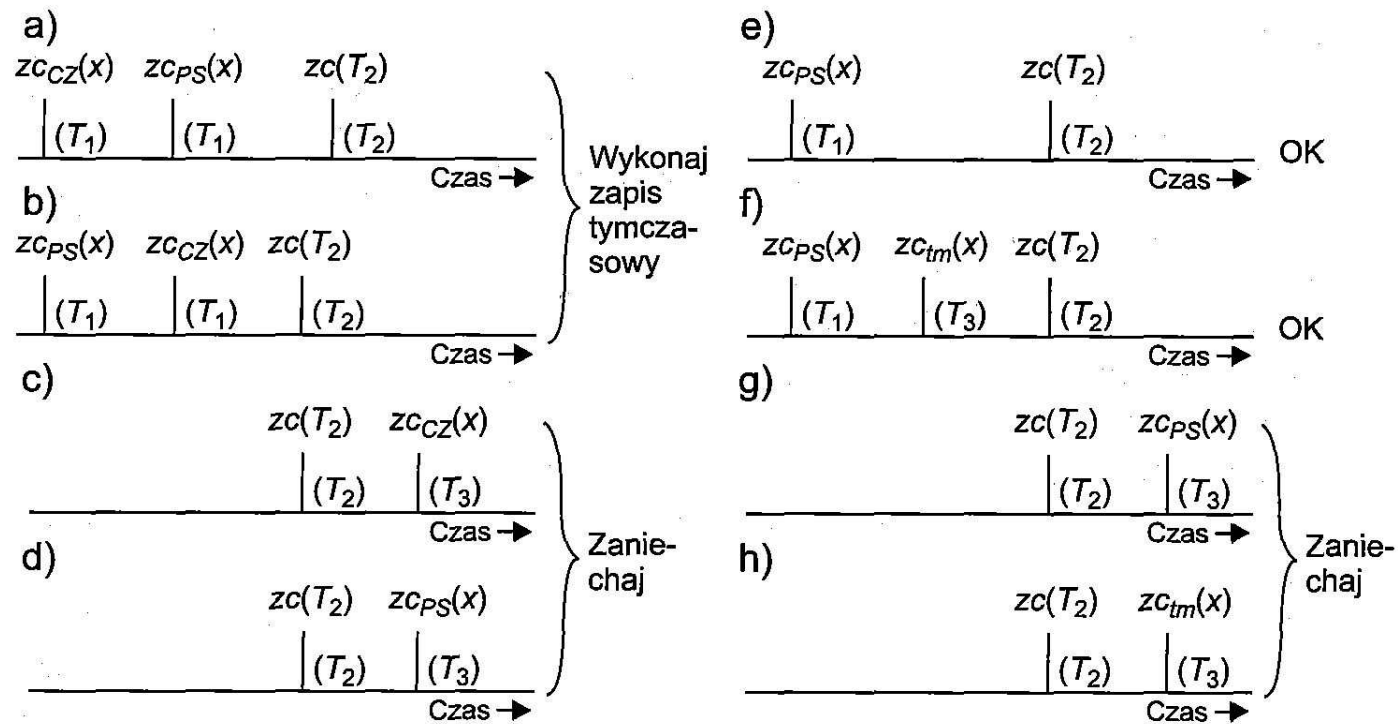
Uwagi:

Stosowanie znaczników powoduje odmienne postępowanie niż w przypadku metody blokowania. Transakcja, która spotka późniejszy znacznik musi być zaniechana.

Stosowanie znaczników jest bezpieczniejsze od metod blokowania - zapobiega powstawaniu zakleszczeń.

Przykład

Trzy transakcje T_1 , T_2 , T_3 ze znacznikami czasu rozpoczęcia zc , czytania zc_{cz} , pisania zc_{ps}



Sytuacje:

- a i b: późniejszy znacznik czasu transakcji T_2 umożliwia jej dokonanie tymczasowego zapisu,
- c i d: działania późniejszej transakcji T_3 powodują zaniechanie transakcji T_2 , która chciała dokonać zapisu,
- e i f: późniejszy znacznik czasu transakcji T_2 umożliwia jej dokonanie czytania,
- g i h: działania późniejszej transakcji T_3 powodują zaniechanie transakcji T_2 , która chciała dokonać odczytu.

BLOKADY W SYSTEMACH ROZPROSZONYCH

Metody postępowania - analogiczne jak w przypadku systemów jednoprocessorowych:

- . Zapobieganie.
- . Unikanie.
- . Wykrywanie i usuwanie skutków.
- . Ignorowanie problemu.

Przykłady postępowania

1. Scentralizowane wykrywanie blokady

System: zbiór maszyn, jeden proces - koordynator.

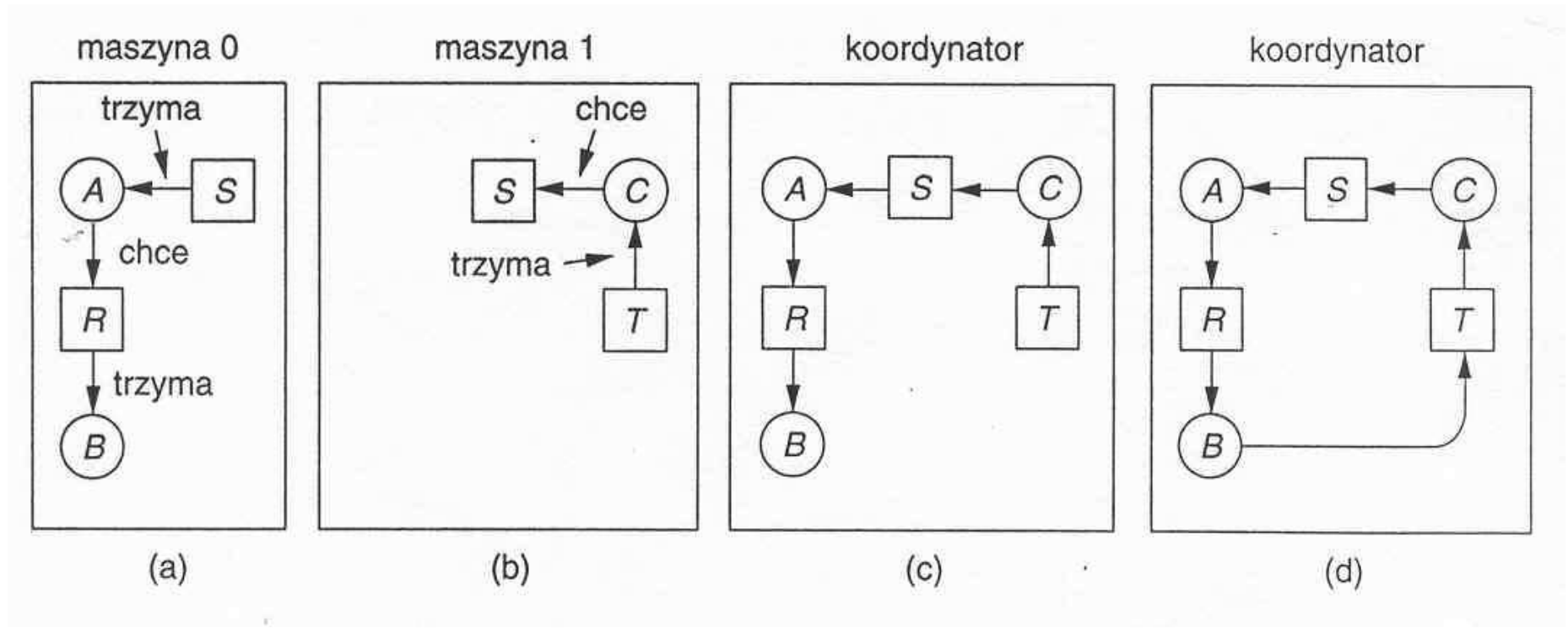
Każda maszyna utrzymuje graf własnych procesów i zasobów.

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.

Przykład ilustrujący możliwość powstania tzw. fałszywej blokady:



Proces B zwolnił zasób R i zamówił zasób T. Komunikat o zwolnieniu zasobu nie dotarł na czas do koordynatora. Koordynator stwierdza cykl w grafie.

Poprawne rozwiązanie:

Logiczna synchronizacja czasu (algorytm Lamporta).

Komunikaty zawierają znaczniki czasu.

Koordynator wykorzystując znaczniki czasu może wyeliminować tworzenie fałszywych blokad.

2. Rozproszone wykrywanie blokady

Przykład algorytmu Chandy-Misra- Haasa.

Proces może zamawiać wiele zasobów jednocześnie.

Idea:

Proces oczekujący na zasób wysyła komunikat do procesu przetrzymującego ten zasób.

Komunikat zawiera:

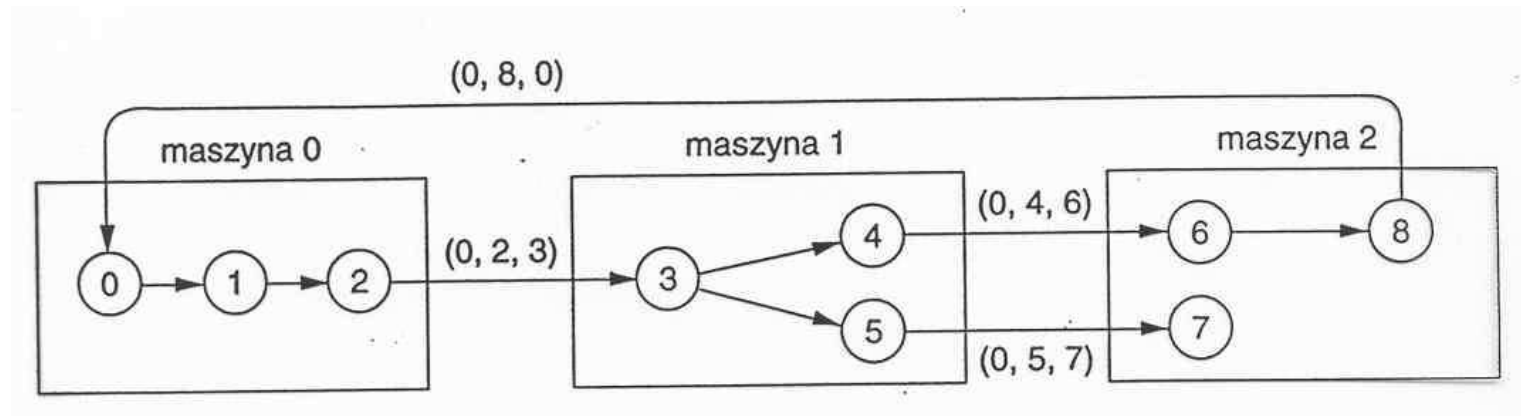
- Nr procesu rozpoczynającego czekanie.
- Nr procesu wysyłającego komunikat.
- Nr procesu, do którego komunikat jest wysyłany.

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.

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

Na rysunku ilustracja dla 3 maszyn (tylko grafy oczekiwań).



3. Zapobieganie blokadom w systemach rozproszonych

przykład praktycznego podejścia wykorzystującego niepodzielne transakcje i znaczniki czasu

Każda transakcja ma jednoznacznie przyporządkowany niepowtarzalny znacznik czasu.

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.

Własność algorytmu: znaczniki procesów oczekujących w łańcuchu są rosnące, zapobiega to powstaniu cyklu.