

PROBLEMY SYNCHRONIZACJI CZASU

Cechy algorytmów rozproszonych:

- Informacje rozmieszczone na wielu maszynach
- Procesy podejmują decyzje tylko na podstawie informacji lokalnych
- Należy unikać skupiania elementów wrażliwych na awarie w jednym punkcie systemu
- W systemie rozproszonym nie istnieje jedno wspólne precyzyjne źródło czasu (wspólny zegar)

SYNCHRONIZACJA ZEGARÓW

Ilustracja działania programu `make` w systemie dwóch maszyn o niezależnych zegarach.

Logiczna synchronizacja zegarów

Zegary logiczne (logical clocks): zapewniające wewnętrzną zgodność czasu.

Fizyczna synchronizacja zegarów

Zegary fizyczne (physical clocks): czas wskazywany przez zegary jest zgodny z czasem rzeczywistym (z określoną dokładnością)

Algorytm synchronizacji zegarów logicznych (Lamport)

Rozpatrujemy system rozproszony, w którym jest wiele procesów, każdy na innej maszynie, każdy ma własny czasomierz.

Relacja uprzedniości zdarzeń (happens-before relation):

Def. Mówimy, że zdarzenie a poprzedza zdarzenie b i piszemy $a \rightarrow b$, wtedy i tylko wtedy, gdy wszystkie procesy są zgodne co do tego, że zdarzenie a zachodzi najpierw, a potem dopiero zdarzenie b .

Relacja ta zachodzi bezpośrednio w przypadkach:

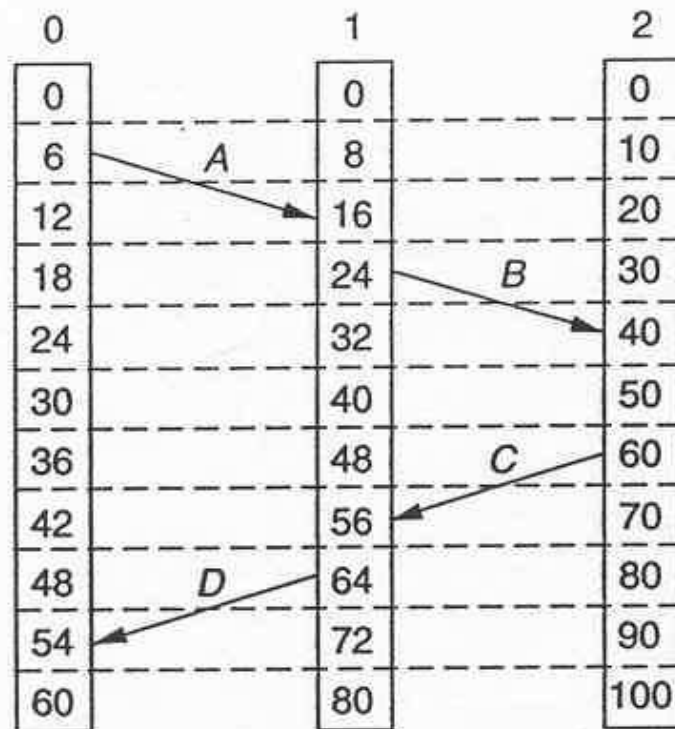
1. Jeżeli a i b są zdarzeniami w tym samym procesie i a występuje przed b , to relacja $a \rightarrow b$ jest prawdziwa.
2. Jeżeli a jest zdarzeniem wysłania komunikatu przez jeden proces i b jest zdarzeniem odebrania komunikatu przez inny proces, to relacja $a \rightarrow b$ jest prawdziwa.

Przechodność relacji: jeżeli $a \rightarrow b$ i $b \rightarrow c$, to $a \rightarrow c$.

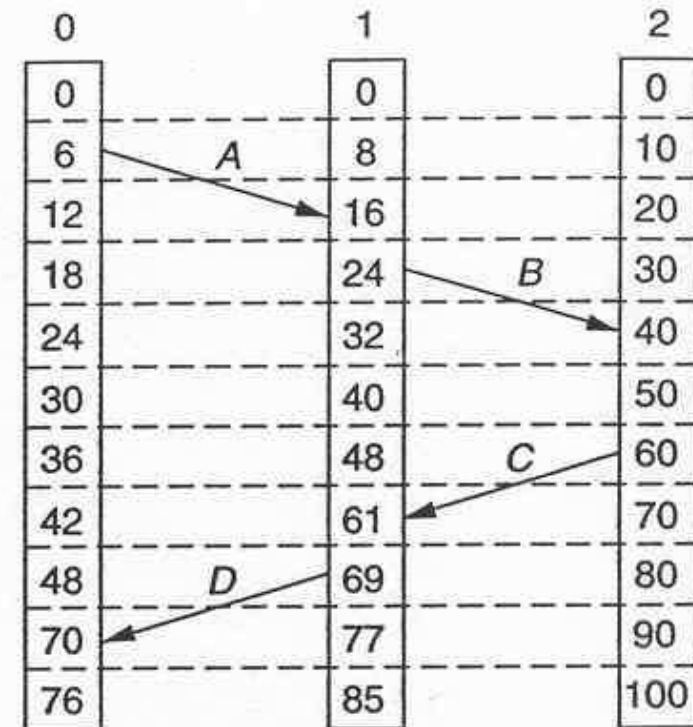
Zdarzenia współbieżne (concurrent): dwa zdarzenia występujące w różnych procesach, które nie wymieniają komunikatów.

Przypisanie wartości czasu zdarzeniom (ozn. $C(a)$) powinno mieć własności:

1. Jeżeli $a \rightarrow b$ to $C(a) < C(b)$.
2. Czas zegarowy C musi zawsze wzrastać.



(a)



(b)

Rys. Trzy procesy (0, 1, 2) z własnymi zegarami bez korekty (a) i z korektą (b) wg. algorytmu Lamporta.

Warunki przypisania czasu wszystkim zdarzeniom w systemie rozproszonym
(zastosowane w algorytmie Lamporta)

1. Jeżeli zdarzenie a poprzedza zdarzenie b w tym samym procesie, to $C(a) < C(b)$.
2. Jeżeli a oznacza nadanie komunikatu, a b jego odebranie, to $C(a) < C(b)$.
3. Dla wszystkich zdarzeń a i b , $C(a) \neq C(b)$.

SYNCHRONIZACJA ZEGAROW FIZYCZNYCH

wymagana, gdy czas przypisywany zdarzeniom w systemie rozproszonym powinien się pokrywać z czasem rzeczywistym (np. w systemach czasu rzeczywistego).

Problemy pomiaru czasu

Czas astronomiczny

sekunda słoneczna: 1/86400 dnia słonecznego (między górowaniami słońca)
średnia sekunda słoneczna (mean solar second)

Międzynarodowy czas atomowy - TAI (International Atomic Time)

czas określonej liczby przejść atomu cezu 133

Bureau International de l'Heure w Paryżu podaje średnią z zegarów atomowych z ok. 50 laboratoriów.

Uniwersalny czas skoordynowany - UTC (Universal Coordinated Time)

Czas atomowy skoordynowany z czasem astronomicznym przez dodawanie sekund przestępnych.

Wzorzec dla wszystkich współczesnych cywilnych pomiarów czasu udostępniany przez:

NIST (National Institute of Standard Time), Fort Collins, Colorado nadajnik krótkofalowy **WWV** i inne stacje radiowe.

Satelitę GEOS (Geostationary Environment Operational Satellite) i inne satelity.

Przykład scentralizowanego algorytmu synchronizacji

Założenia

System rozproszony - wiele maszyn, jedna (serwer czasu) ma odbiornik WWV.

Zakłada się, że każda maszyna ma czasomierz powodujący H przerwań na sek. Czas liczony jest jako liczba impulsów zliczanych od pewnej ustalonej chwili w przeszłości (kolejny impuls dodawany jest w momencie kolejnego wyzerowania czasomierza).

Maksymalny współczynnik odchylenia (maximum drift time): stała ρ taka, że

$$1 - \rho \leq dC/dt \leq 1 + \rho,$$

gdzie

$C(t)$ - czas wskazywany przez zegar maszyny względem czasu teoretycznego.
(Rzeczywista liczba przerwań na sek. w różnych maszynach może odbiegać od H).

Zapewnienie odchylenia czasu między dwiema maszynami nie większego niż δ wymaga korekty zegarów nie rzadziej niż co $\delta / (2 \rho)$ sekund.

Idea algorytmu synchronizacji czasu fizycznego (Cristian)

Każda maszyna okresowo (co $\delta / (2 \rho)$ sekund) wysyła komunikat do serwera czasu z pytaniem o bieżący czas.

Serwer czasu podaje w odpowiedzi czas UTC ozn. C_{UTC} .

Każda z maszyn koryguje czas (stopniowo).

Należy uwzględnić czas przenoszenia komunikatu.

WZAJEMNE WYŁĄCZANIE W SYSTEMACH ROZPROSZONYCH

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

Algorytm scentralizowany

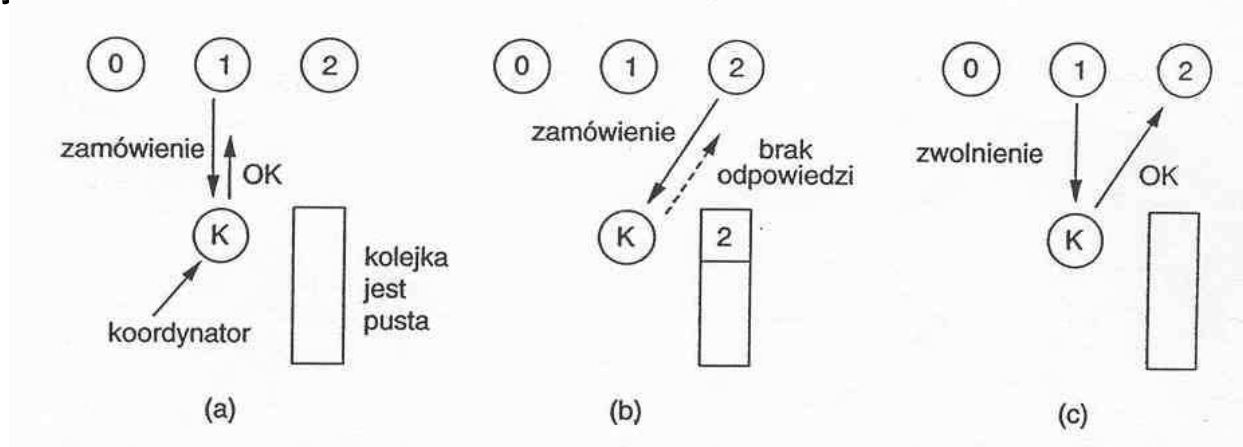
Jeden proces jest koordynatorem.

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 wyłączenie, nie zachodzi głodzenie procesów, łatwy w realizacji, wrażliwy na awarie.

Algorytm rozproszony

Wymagane jest 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 wyłączenia bez głodzenia.

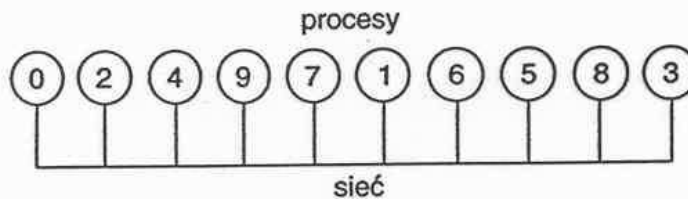
Wrażliwy na awarie - brak odpowiedzi spowodowany awarią procesu jest traktowany jako brak zgody - blokowanie procesów próbujących wejść do sekcji krytycznej (jest możliwość rozwiązania).

Wymagana komunikacja grupowa lub każdy proces musi utrzymywać listę procesów znajdujących się w grupie, wchodzących i wychodzących - obciążenie systemu.

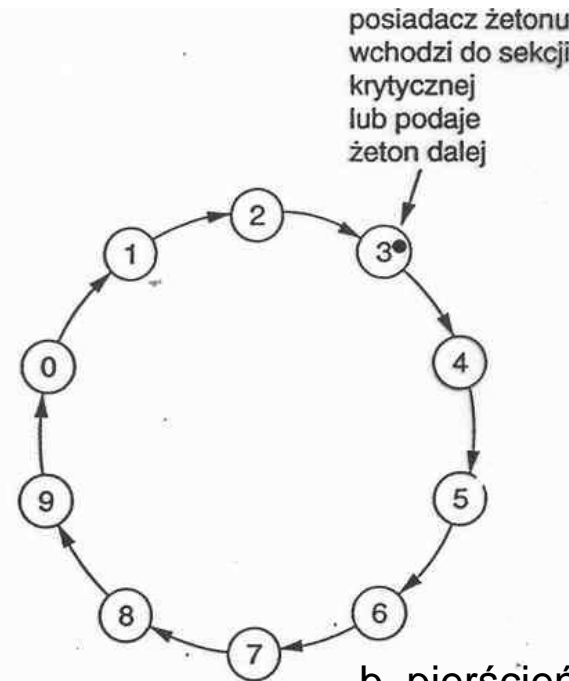
Algorytm pierścienia logicznego z żetonem

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 - zatrzym



a. nieuporządkowana grupa procesów w sieci fizycznej.



b. pierścień logiczny zbudowany za pomocą oprogramowania.

Cechy algorytmu:

Zapewnia wzajemne wyłączenie. Nie powoduje głodzenia procesów.

Powstają problemy związane z zaginięciem żetonu.

Wrażliwy 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,

próbują się zlokalizować proces o największym numerze.

Algorytm tyrana:

1. Proces A zauważył, że koordynator nie odpowiada.

A wysłała komunikat ELEKCJA do wszystkich procesów z większymi numerami.

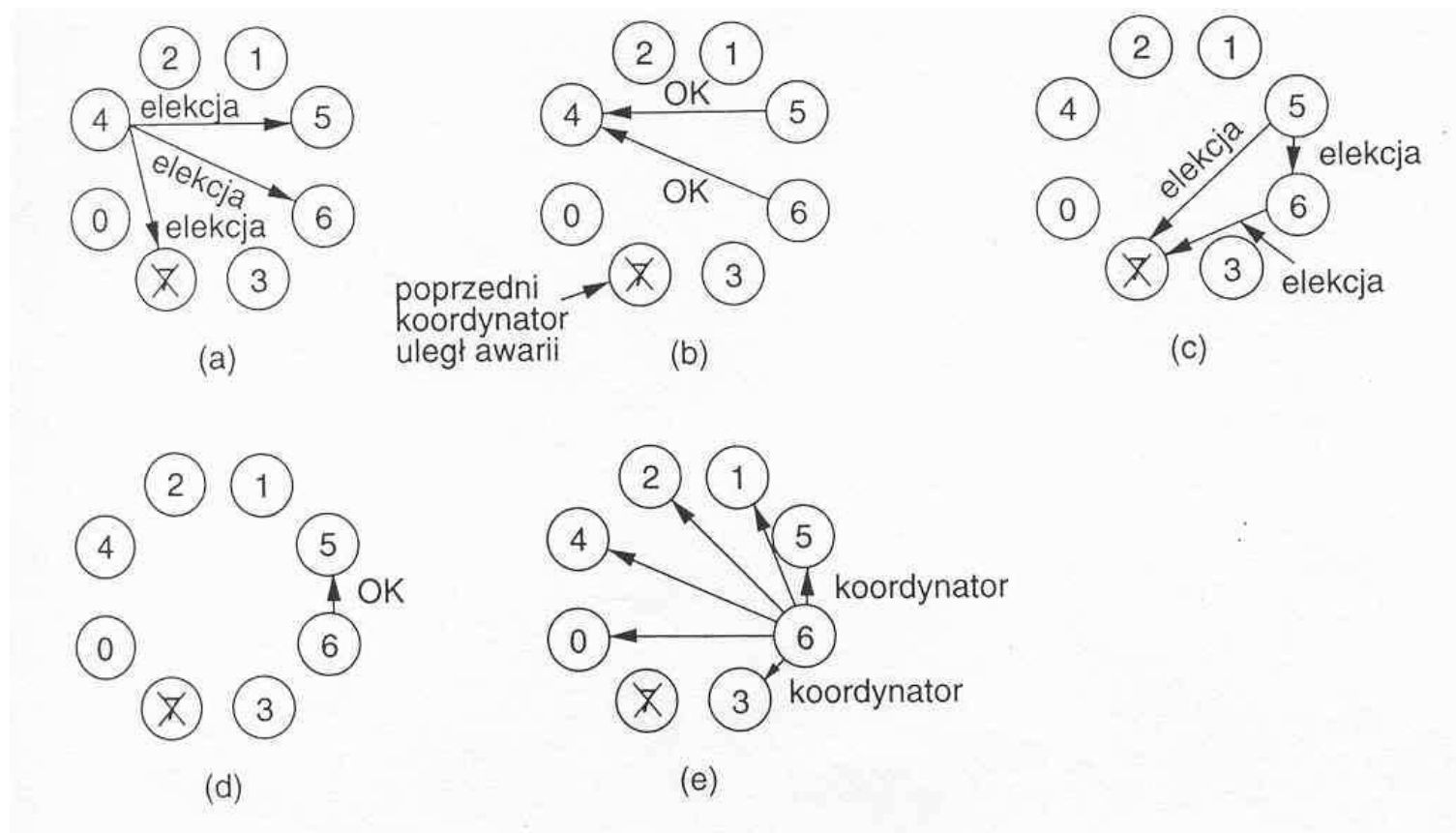
2. Brak odpowiedzi, to A zostaje koordynatorem.

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 wysła 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 wyłączzonego 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, który zostaje koordynatorem.
4. Koordynator rozpoczyna działanie.

