część I (L. Kruś)

WYŻSZA SZKOŁA INFORMATYKI STOSOWANEJ
I ZARZĄDZANIA
WYDZIAŁ INFORMATYKI

ROZPROSZONE SYSTEMY OPERACYJNE

KONSPEKT WYKŁADÓW

STUDIA DZIENNE, WIECZOROWE, ZAOCZNE
Semestr 5, Rok: 2001/2002

Wykładowcy: Lech Kruś, Mikołaj Aleksiejuk

Cel przedmiotu:

Wprowadzenie słuchaczy do współczesnych zagadnień systemów rozproszonych i sieci komputerowych.

Ułatwienie rozumienia podstawowych zagadnień związanych z komunikacją w sieciach komputerowych i działaniem systemów rozproszonych, w tym, takich jak: wielowarstwowe protokoły komunikacji w sieciach, działanie w układzie klient serwer, adresowanie w internecie, zdalne wykonywanie prac, przekazywanie komunikatów, synchronizacja i zarządzanie procesami, rozproszone systemy plików, zagadnienia tolerowania awarii, wprowadzenie do zarządzania w sieciach.

Uzupełnieniem wykładu są zajęcia laboratoryjne poświęcone zagadnieniom sieci komputerowych

Wymagane przygotowanie słuchaczy:

- w zakresie podstaw informatyki technicznej (cyfrowej reprezentacji informacji, podstaw arytmetyki komputerów, podstaw teorii układów logicznych: A. Skorupski, Podstawy budowy i działania komputerów, WKŁ 1996),
- organizacji i architektury komputerów (P. Metzger, Anatomia PC, Helion 1996),
- wielodostępnych systemów operacyjnych I (wykłady i laboratorium na Wydz. Informatyki, WSISiZ)
- wielodostępnych systemów operacyjnych II - zagadn. zaawansowane (wykłady i laboratorium na Wydz. Informatyki, WSISiZ)

Zaliczenie przedmiotu: zaliczenie laboratorium (kolokwium), egzamin

jeśli ktoś będzie miał 5 ma zaliczony egzamin

Zakres tematyczny wykładów:

I. Zagadnienia budowy systemów rozproszonych

Synchronizacja w systemach rozproszonych
Synchronizacja czasu logicznego i czasu fizycznego
Algorytmy synchronizacji procesów
Algorytmy elekcji

Transakcje niepodzielne
Założenia przetwarzania transakcyjnego
Metody realizacji: prywatna przestrzeń robocza,
Protokół zatwierdzania dwufazowego
Protokoły współbieżnego przetwarzania transakcji

Blokady w systemach rozproszonych
Algorytm scentralizowanego rozpoznawania blokady
Algorytm zdecentralizowany

Procesy i procesory w systemach rozproszonych
Praca wielowątkowa
Synchronizacja wątków
Modele systemów
Model stacji roboczej
Model puli procesorów

Rozproszone systemy plików
System NFS

Zagadnienia tolerowania awarii
Wady elementów systemu
Awarie systemu
Redundancja
Zwielokrotnienie aktywne
Zasoby rezerwowe
Uzgodnienia w systemach wadliwych
Przykład systemu MC Service Guard firmy Hewlett-Packard

Wprowadzenie do zarządzania w sieciach komputerowych
Model zarządzania: stacja zarządzająca – „agent" na stacji roboczej
Zmienne wykorzystywane do zarządzania
Baza MIB

WYKŁAD I

Przykład rodziny produktów Open View firmy Hewlett Packard

Rozproszone środowisko obliczeniowe (DCE)

II. Podstawy sieci komputerowych

Praca w intersieci (internetworking)
- Bazowe techniki sieciowe (Ethernet, FDDI, ATM)
- Specyfikacja Ethernetu
- Metody dostępu do sieci (CSMA/CD, znacznika)
- Budowa ramek (Ethernet, FDDI)
- Adresy logiczne i fizyczne

Łączenie sieci w intersieć i model jej architektury
- Przyporządkowywanie adresowi logicznemu adresu fizycznego (ARP)
- Protokół intersieciowy (IP)
- Budowa datagramu

- Protokół ICMP
- Wykorzystanie ICMP – Ping
- Protokół TCP
- Komunikacja TCP z górnymi warstwami
- Porty, TCP i połączenia
- Budowa segmentu TCP
- Protokół UDP

Model warstwowy oprogramowania protokołów
- 7- warstwowy model ISO
- Podział na warstwy w środowisku sieciowym TCP/IP

Programy użytkowe
- Programy użytkowe do pracy na odległym komputerze: telnet, rlogin
- Przesyłanie plików i dostęp : FTP, TFTP, NFS
- Poczta elektroniczna: SMTP

Konfiguracja i podstawy administrowania TCP/IP
- Pliki konfiguracyjne
- Demon inetd
- Polecenie netstat

Obsługa nazw domenowych (DNS)
- Struktura DNS
- Rekordy zasobów
- Komunikaty DNS

Interfejs gniazd
- Sieciowe wejście-wyjście, pojęcie gniazd
- Wysyłanie i odbieranie danych przez gniazdo
- Zdalne wywoływanie procedury (RPC)
- Maper portów

Przyszłość TCP/IP
- Protokół IPv6
- Technologie szybkich sieci lokalnych

Dobre to przeczytać

Literatura podstawowa:

Andrew S. Tanenbaum: Rozproszone systemy operacyjne. PWN. Warszawa 1997.

G. Coulouris, J. Dollimore, T. Kindberg : Systemy rozproszone, podstawy i projektowanie. WNT, Warszawa, 1998

Douglas E. Comer: Sieci komputerowe TCP/IP. (Tom 1) Zasady, protokoły i architektura. WNT, Warszawa 1997.

Larry L. Peterson, Bruce S. Davie: Sieci komputerowe – podejście systemowe. Nakom, Poznań.2000.

Literatura uzupełniająca:

W. Richard Stevens: TCP/IP Illustrated Volumel Protocols, Addison Wesley, New York, 1994

Craig Hunt: TCP/IP Administracja Sieci. O'Reilly&Associates Inc., Oficyna Wyd. READ ME, Warszawa, 1996.

Abraham Silberschatz, James.L. Peterson, Peter.B. Galvin; Podstawy systemów operacyjnych, WNT, Warszawa 1993.

Maurice. J. Bach; Budowa systemu operacyjnego UNIX, WNT, Warszawa, 1995.

Andrew. S. Tanenbaum; Modern Operating Systems, Prentice-Hall, Inc. London, 1992.

Abraham Silberschatz, Peter.B. Galvin; Operating System Concepts. Addison Wesley, New York, 1994

Douglas E. Comer, D. L. Stevens: Sieci komputerowe TCP/IP. (Tom 2) Projektowanie i realizacja protokołów. WNT, Warszawa 1997.

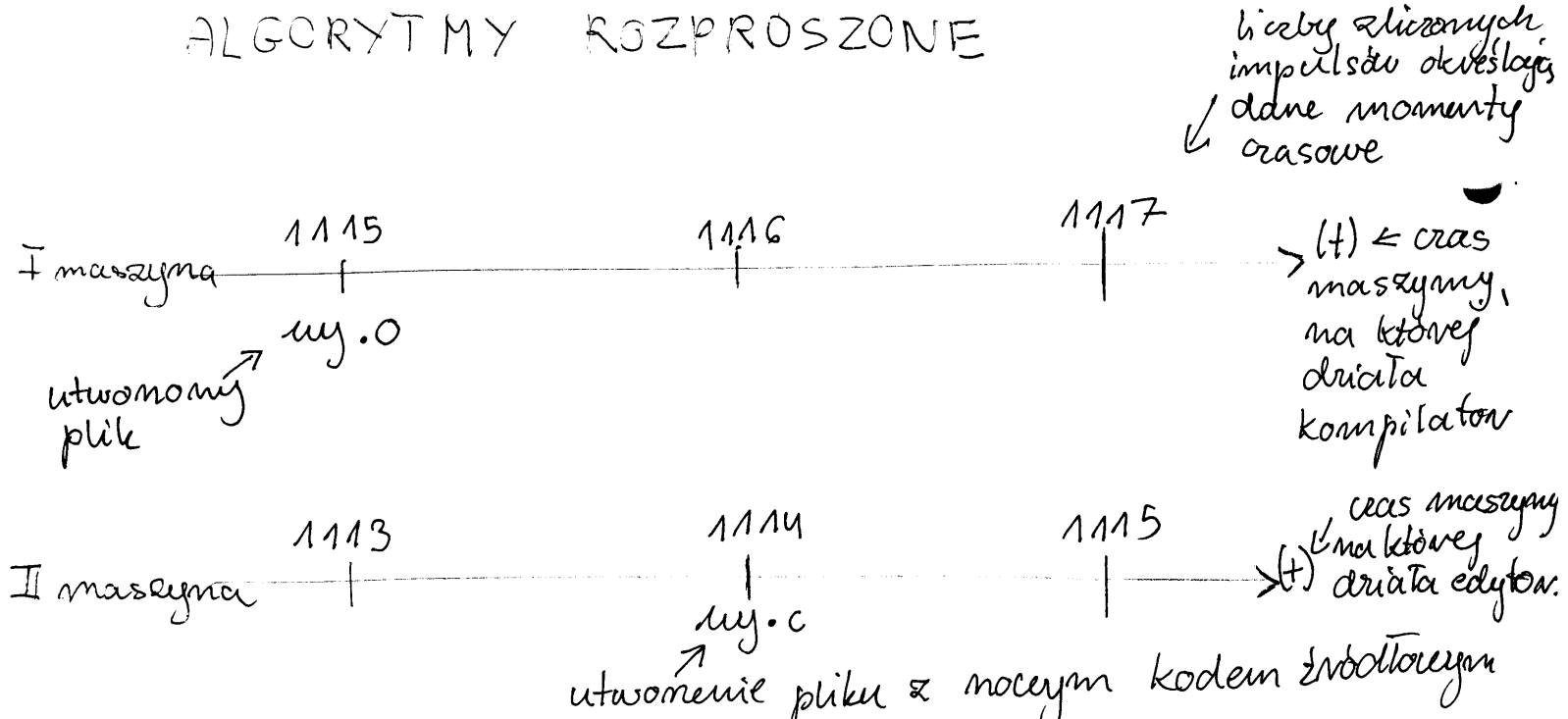
Douglas E. Comer, D. L. Stevens: Sieci komputerowe TCP/IP. (Tom 3) Programowanie w trybie klient serwer. Wersja BSD. WNT, Warszawa 1997.

Simson Garfinkel, Gene Spafford. Bezpieczeństwo w Unixie i Internecie. Wydawnictwo RM i O'Reilly& Associates, Inc. Warszawa 1997.

Bruce Schneier: Kryptografia dla praktyków. WNT i John Wiley&Sons Inc. Warszawa 1995.

Bruce Schneier: Ochrona poczty elektronicznej. WNT i John Wiley&Sons Inc. Warszawa 1996.

ALGORYTMY ROZPROSZONE



Programista używa dwóch maszyn z oddzielnymi zegarami. Jak napisze program to uruchamia MAKE, który sprawdza znaczniki czasowe, jeśli stwierdzi że plik wy.c ma wcześniejszy czas to nie skompiluje nowego kodu źródłowego, który powstał później niż plik wy.o, ale ma wcześniejszy znacznik czasowy. Zależne jest to od zegarów maszyn, które nie zawsze mogą takie same czasy.

SYNCHRONIZACJA ZEGARÓW

Zegar składa się ze stabilizatora czasowego, licznika i rejestrów pretencyjnych. Czas określany jest liczbą impulsów liczonych od pewnych momentów z przeszłości. Dwa zegary mogą generować różną liczbę impulsów. Następują wtedy odchylenia czasowe impulsów, powodując że dwie maszyny mogą mieć różne czasy.
Nie jest konieczna dokładność zegarów z czasem astronomicznym, ale znaczniki czasowe powinny mieć określoną kolejność; najpierw zdarzenie wcześniejsze, a potem zdarzenie późniejsze z późniejszym znacznikiem czasowym – SYNCHRONIZACJA LOGICZNA. Znaczniki czasu powinny odzwierciedlać faktyczną kolejność zdarzeń.

SYNCHRONIZACJA PROCESÓW W SYSTEMACH ROZPROSZONYCH

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, żeby uniknąć braku działania systemu
- Nie istnieje wspólne precyzyjne źródło czasu (wspólny zegar), bo każda maszyna ma swój zegar i trzeba je zsynchronizować.

SYNCHRONIZACJA ZEGARÓW

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

Logiczna synchronizacja zegarów

Zegary logiczne (logical clocks): zapewniające wewnętrzną zgodność czasu
Zegary fizyczne (physical clocks): czas wskazywany przez zegary jest zgodny z czasem rzeczywistym (z określoną dokładnością) np. samoloty
Tak trzeba je synchronizować, żeby znaczniki czasowe były zgodne z czasem rzeczywistym.

Algorytm synchronizacji zegarów logicznych (Lamport)

Rozpatrujemy system rozproszony, w którym jest wiele procesów, każdy na innej maszynie, każdy ma własny czasomierz. są połączone siecią, są tam różne procesy

Relacja uprzedniości zdarzeń (happens-before relation):
Def. : Mówimy, że zdarzenie **a** poprzedza zdarzenie **b** i piszemy
a -> b
wtedy i tylko wtedy, gdy wszystkie procesy są zgodne co do tego, że zdarzenie **a** zachodzi najpierw, a potem dopiero zdarzenie **b**.
procesy na różnych maszynach z różnymi zegarami

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 -> 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 -> b** jest prawdziwa. (odbiór nie może nastąpić przed nadaniem.

Przechodniość relacji: jeżeli a -> b i b -> c, to a -> c.

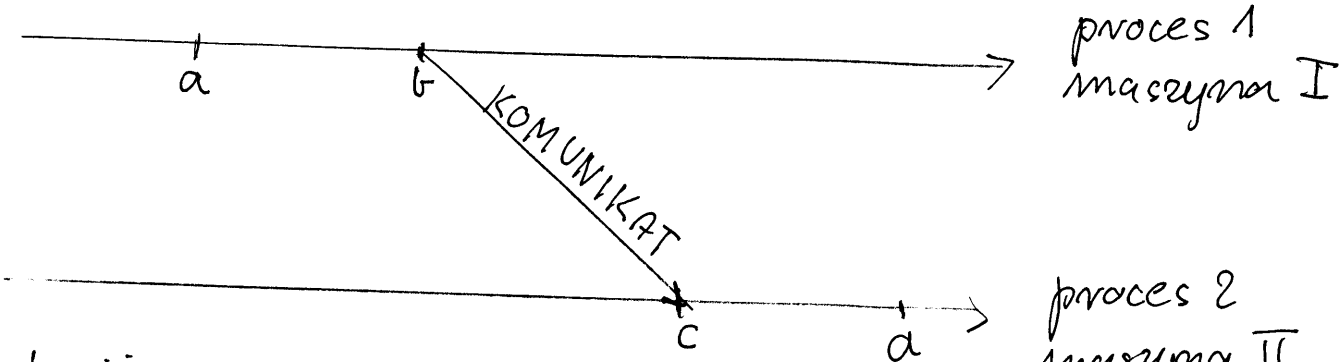
Zdarzenia współbieżne (concurrent): dwa zdarzenie występują w różnych procesach, które nie wymieniają komunikatów.

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

1. **Jeżeli a -> b to C(a) < C(b).**
2. **Czas zegarowy C musi zawsze wzrastać.** - jeśli dokonujemy korekty czasu w systemie konkurencyjnym to możemy tylko dodawać impulsy.

Algorytm synchronizacji zegarów logicznych:

Na I maszynie jest proces 1, na II maszynie proces 2, na III maszynie proces 3. Każda maszyna ma inny czas



Z relacji przechodniości d jest późniejsze od a, b i c.

Jeśli procesy nie wymieniają komunikatów to nie wiemy jak zdarzenia są ze sobą powiązane - mówimy wtedy o zdarzeniach współbieżnych

Czas - przypisanie pewnej liczby impulsów zdarzeniom. Zdarzenie późniejsze ma większy znacznik czasu.

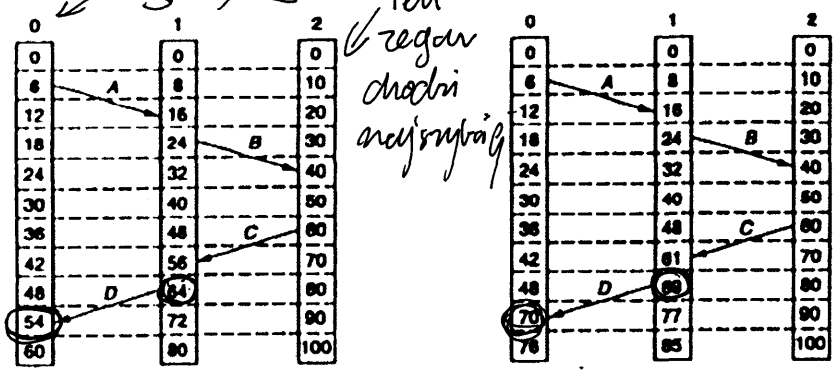
ALGORYTM LAMPARTA:

1. Komunikat zawiera czas nadania (znacznik czasu)
2. Proces odbierający komunikat porównuje czas nadania ze swoim czasem odbioru.
   - Jeśli czas odbioru jest późniejszy nie dokonuje żadnej korekty.
   - Jeśli czas odbioru jest taki sam lub wcześniejszy wtedy dokonuje korekty, dodając tyle impulsów, aby czas odbioru był o 1 impuls późniejszy od czasu nadania. Korekta dokonywana jest na maszynie procesu odbierającego.

Po dodaniu impulsów znaczniki czasu są późniejsze o określoną ilość impulsów (są przesunięte do przodu)

a) sytuacja przed synchronizacją zegarów

b) Po korekcie wg algorytmu Lamporta



A, B, C i D → komunikaty

Dodajemy 16 impulsów

Rys. Trzy procesy (0, 1, 2) z własnymi zegarami bez korekty i z korektą 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) ≠ C(b).

↑ znaczniki czasowe a

Różne zdarzenia muszą mieć różne znaczniki czasu

SYNCHRONIZACJA ZEGARÓW FIZYCZNYCH

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

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 stabilny, nie uwzględnia fluktuacji
czas określonej liczby przejść atomu cezu 133 Ziemi
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,
Satelitę **GEOS** (Geostationary Environment Operational Satellite).

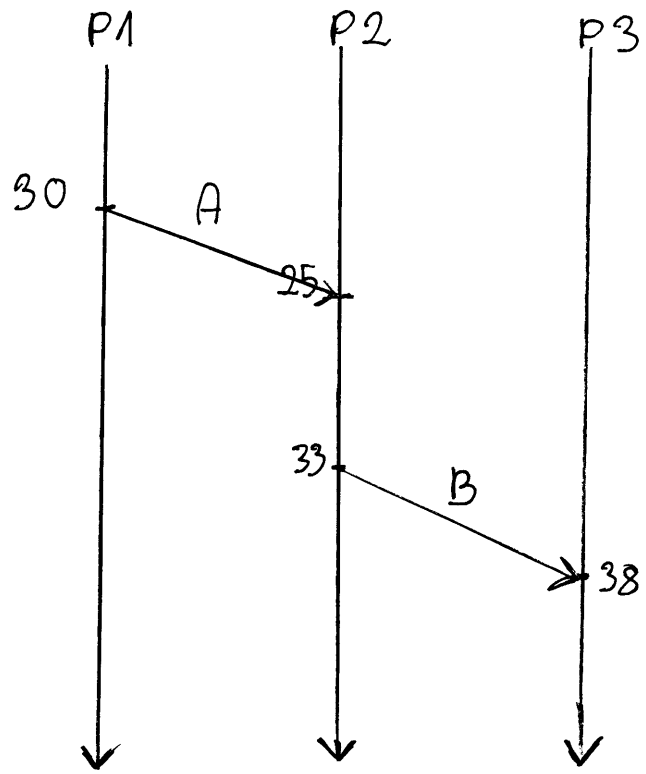
Co około 260 dni dodaje się 1 sekundę przestępną

ZADANIE

Należy rozważyć trzy procesy, każdy działający na innej maszynie. Każda maszyna ma lokalny czasomierz. W przypadku bez korekty czasu maszyn proces P1 wysyła komunikat A w chwili 30. Komunikat ten odbierany jest przez proces P2 w chwili 25. Następnie proces P2 w chwili 33 wysyła komunikat B do procesu P3. Proces P3 odbiera ten komunikat w chwili 38. Jakie będą lokalne czasy nadania komunikatu przez proces P3 po dokonaniu synchronizacji logicznej czasu wymienionych maszyn zgodnie z algorytmem Lamporta?

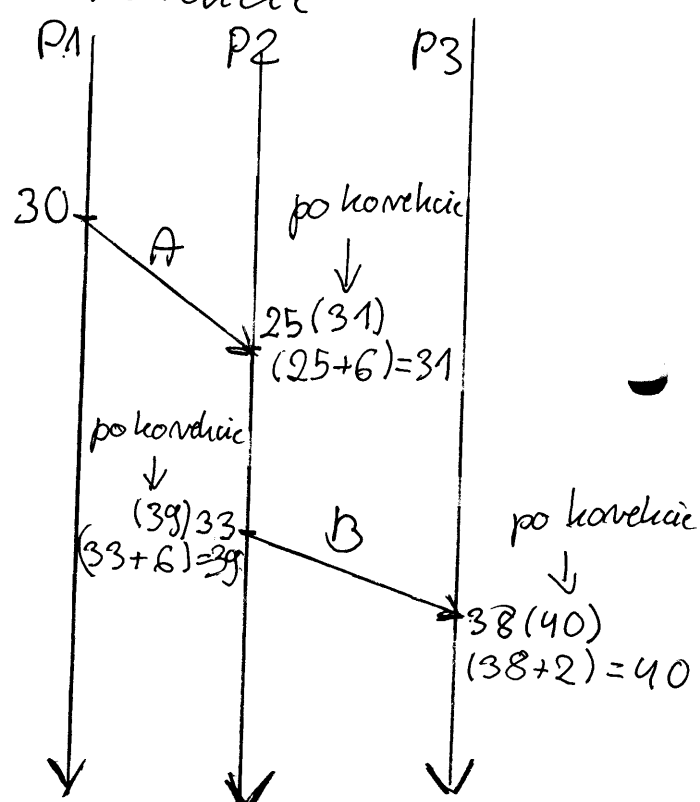
ROZWIĄZANIE

Przed korektą



Czasy odebrania komunikatów A i B są wcześniejsze niż czasy ich wysłania

Po korekcie



Korekta została wykonana na maszynie II i III, tak aby czasy odebrania komunikatów A i B były późniejsze niż czasy ich wysłania.

Przykład scentralizowanego algorytmu synchronizacji. Zegarów fizycznych

Założenia

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

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). Producent określa dokładność swojego zegara

Maksymalny współczynnik odchylenia (maximum drift time):

stała ρ taka, że

$1 - \rho \le dC/dt \le 1+\rho$, 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 δ/2ρ sekund.

Idea algorytmu (Cristian)

Każda maszyna okresowo (co δ/2ρ 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).

Przykłady wykorzystania synchronizacji zegarów

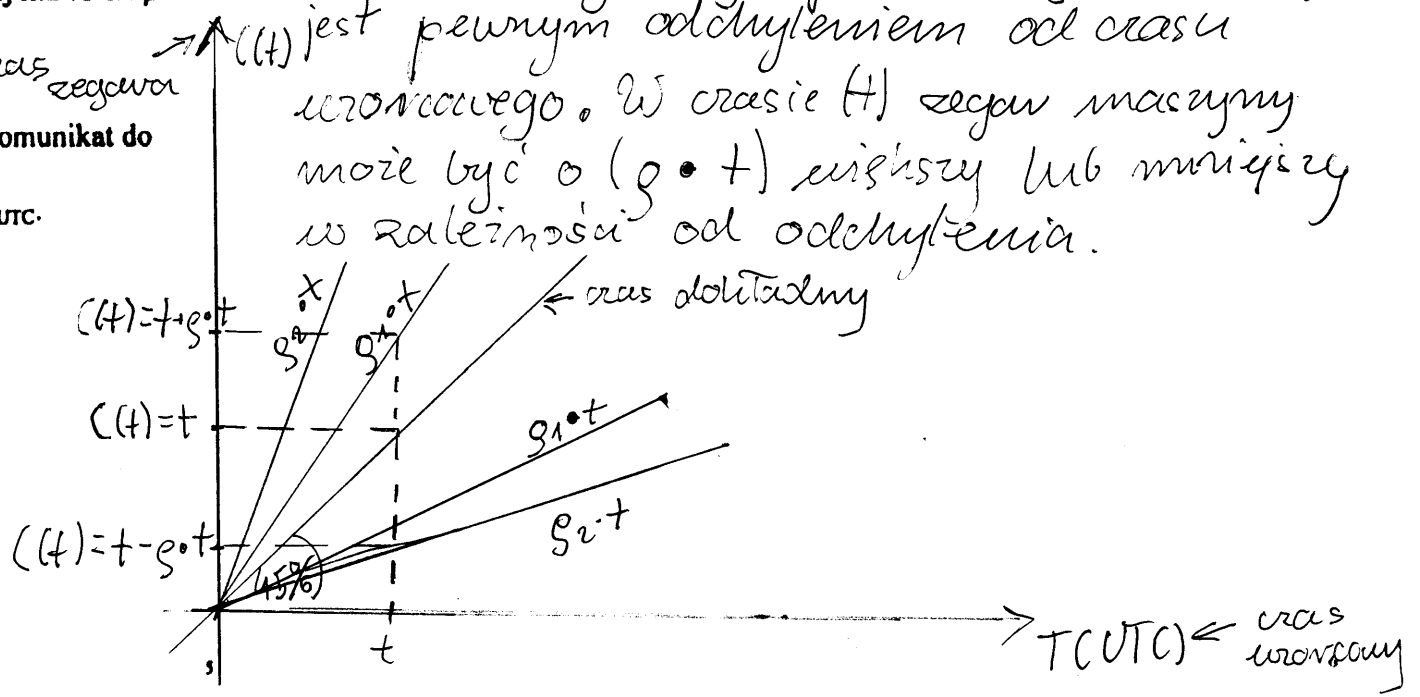
Zapewnienie jednokrotnego doręczania komunikatów (w przypadku awarii serwera).

Zapewnienie spójności pamięci podręcznych.

SYNCHRONIZACJA ZEGARÓW FIZYCZNYCH

Czas fizyczny (astronomiczny) – bieżący pewną liczbę dni, doba ma 86400s. Wyliczanie czasu z milisekundy dzienne w związku z fluktuacją obrotu ziemi. To źródło czasu jest niestabilne

Czas odczytywany przez zegar maszyny jest pewnym odchyleniem od czasu uniwersalnego. W czasie (t) zegar maszyny może być o $(\rho \cdot t)$ większy lub mniejszy w zależności od odchylenia.



1. Dwie maszyny o tym samym ϱ

Najgorszy przypadek, gdy zegar jednej maszyny się przyśpieszy o ϱ, a drugi zwolni o ϱ.

δ - (delta) - odchylenie czasu między dwoma maszynami. (trzeba zastosować korektę)

$$\delta \geq t \cdot 2\varrho \qquad t \leq \frac{\delta}{2\varrho}$$

δ - odchylenie czasu między maszynami

2. Dwie maszyny z ϱ_1 i ϱ_2

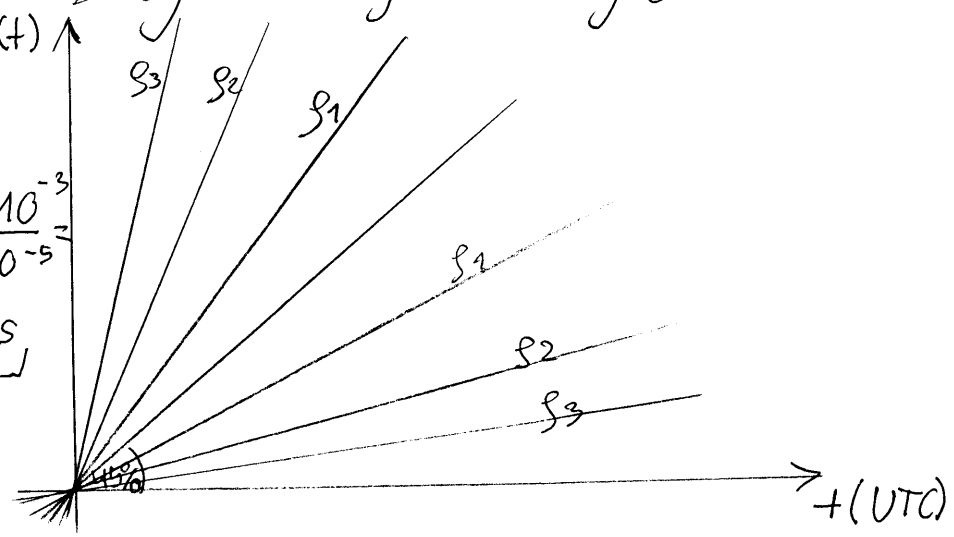
$$\delta \geq t \cdot (\varrho_1 + \varrho_2) \qquad t \leq \frac{\delta}{\varrho_1 + \varrho_2}$$

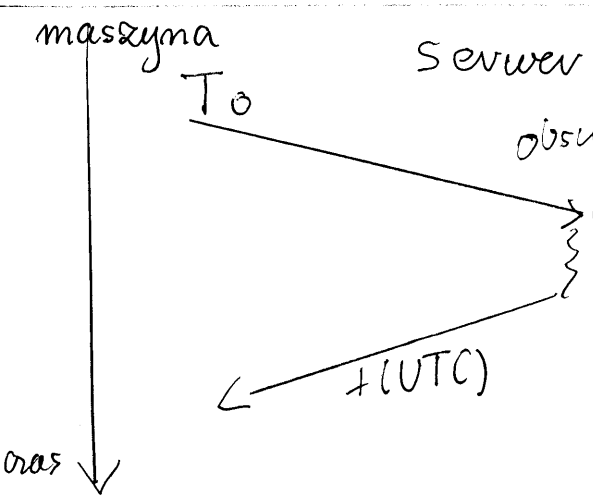
ZADANIE

Mamy trzy maszyny z $\varrho_1 = 15 \cdot 10^{-5}$, $\varrho_2 = 7 \cdot 10^{-5}$, $\varrho_3 = 13 \cdot 10^{-5}$. Co ile sekund synchronizować zegary, aby odchylenie czasu między maszynami było $\leq$ 20 milisekund

Rozwiązanie

$$t \leq \frac{20\,ms}{\varrho_1 + \varrho_3} = \frac{20\,ms}{28 \cdot 10^{-5}} = \frac{20 \cdot 10^{-3}}{28 \cdot 10^{-5}} = \frac{200 \cdot 100^{s}}{28} = \frac{200}{2{,}8}\,s \approx \underline{70\,s}$$





Maszyna w chwili T_1 dokonuje korekty czasu uwzględniając czasy przesyłania komunikatu

Korekty nie można dokonać wstecz, gdyż zegar to częstotliwość generowania impulsów. Należy zwolnić czas maszyny, żeby zgodził się z czasem wzorcowym