

---

# Sisteme distribuite – Teorie

## 6. Sincronizarea ceasurilor - ceasuri logice vs. fizice

---

---

# Sicronizare: sisteme monoprocesor vs. sisteme distribuite

- Mono-procesor:
  - Regiuni critice, excludere mutuala, si alte probleme de sincronizare sunt rezolvate cu ajutorul semafoarelor si monitoarelor
- Sisteme distribuite:
  - Semafoarele si monitoarele nu sunt adecvate pentru ca se bazeaza pe existenta memoriei partajate
  - Problemele care trebuie tratate:
    - Timpul
    - Excluderea mutuala
    - Algoritmi de alegere
    - Tranzactii atomice
    - Impas

---

# Construirea alg. pt. sincronizarea ceasului

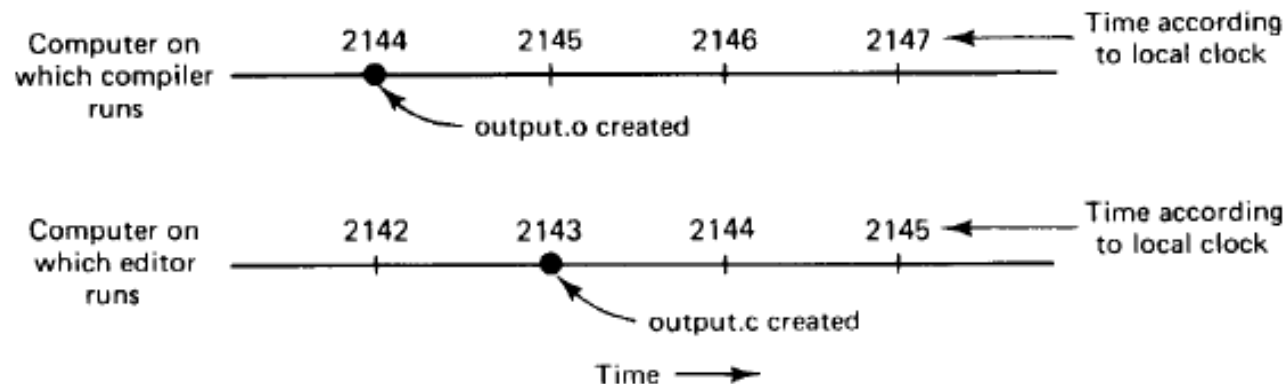
- Abordare centralizata:
  - Colecteaza toata informatia din sistem intr-un singur loc
  - Un proces o examineaza si ia o decizie asa cum se procedeaza in cazul mono-procesor
  - Punct de esec: centralizatorul
  
- Algoritmii distribuiti au urmatoarele proprietati:
  1. Informatia relevanta este stransa la masini multiple
  2. Procesele iau decizii bazate numai pe informatia locala
  3. Un singur punct de esec in sistem trebuie evitat  
De ce? Sist.distr. trebuie sa fie mai de incredere decat masinile individuale  
Daca una cedeaza, restul trebuie sa fie capabil sa continue sa functioneze
  4. Nu exista un ceas comun sau o sursa de timp global precis

---

# Alg distribuiti – a 4<sup>a</sup> proprietate

- Contraex: sistemele centralizate – Intr-un sistem centralizat timpul este ne-ambiguu.
  - Cand un proces doreste sa cunoasca timpul, face un apel de sistem si nucleul ii raspunde
  - Daca procesul A se informeaza asupra timpului, si mai tarziu procesul B se informeaza, valoarea lui B > valoarea lui A.
- Intr-un SD, *ajungerea un acord asupra timpului nu este triviala.*
- Exemplu: lipsa cunoasterii timpului global intr-un cluster de masini Unix.

# Exemplul: make



- Make examinează timpii la care fişierele sursă şi obiect au fost modificate
  - Make urmăreşte toate fişierele sursă pentru a afla care trebuie recompilate şi apelează compilatorul pentru a le recompila
- Normal:
  - Dacă fişierul sursă `input.c` are timpul 2151 şi fişierul obiect corespunzător are timpul 2150, `make` recunoaşte faptul că `input.c` a fost schimbat de când `input.o` a fost creat, şi astfel `input.c` trebuie recompilat.
  - Dacă `output.c` are timpul 2144 şi `output.o` are timpul 2145, nu este necesară o recompilare
- De-sincronizare:
  - La timp scurt după timpul real 2144 `output.c` este modificat pe o maşină cu ceas dat înapoi
  - `Make` nu va apela compilatorul.
  - Programul binar executabil rezultat conţine o mixtură de fişiere obiect de la sursele vechi şi sursele noi

---

# Ceasul calculatorului

- Fiecare calculator are un circuit pentru a tine evidenta timpului
  - Cuvantul "ceas" se refera la acest dispozitiv, dar nu sunt ceasuri in sensul uzual: contor de timp este un cuvant mai adecvat.
- Contorul de timp este bazat pe un cristal de quart prelucrat precis
  - Cand este tinut sub tensiune, cristalul de quart oscileaza la o frecventa bine definita care depinde de tipul de cristal, cum este taiat si cantitatea de tensiune
- Asociat cu fiecare cristal sunt doi registrii, un registru contor si un registru pastrator.
  - Fiecare oscilatie a cristalului decrementeaza contorul cu unu
  - Cand contorul ajunge la zero, este generata o intrerupere si contorul este reincarcat din registrul pastrator
- Astfel este posibila programarea unui contor de timp care genereaza o intrerupere de 60 de ori pe secunda sau la alta frecventa dorita
  - Fiecare intrerupere este numita *tic de ceas*.

---

# Alunecarea ceasului

- Deși frecvența oscilatorului este de obicei stabilă, este imposibilă garantarea faptului că diferite calculatoare vor rula la aceeași frecvență
- Când sistemul are  $n$  calculatoare, cele  $n$  cristale rulează la rate diferite, cauzând ceasurile (software) să iasă gradual din sincronizare și să dea valori diferite la citire.
- Această diferență în valorile de timp este numită *alunecarea ceasului (skew)*.
- Consecința:  
programele care se bazează pe timpul asociat cu un fișier, obiect, proces sau mesaj pot esua – ex. “make”

---

# Timpi relativi

- Problema “Make” in cazul unui singur calculator si a unui singur ceas este rezolvata:
    - Toate procesele pe masina care utilizeaza ceasul vor fi intern consistente
    - Nu conteaza daca ceasul este decalat cu o cantitate mica
    - Tot ceea ce conteaza este *timpul relativ*.
  - Lamport – 1978
    - A demonstrat ca sincronizarea ceasurilor este posibila si a prezentat un algoritm
    - Ideea: sincronizarea ceasurilor nu trebuie sa fie absoluta
      - Daca doua procese nu interactioneaza, nu este necesara sincronizarea ceasurilor lor deoarece lipsa sincronizarii nu va fi observata so nu va cauza probleme
      - Ceea ce conteaza in mod uzual nu este ca procesele sa fie de acord asupra timpului exact, ci sa fie de acord asupra ordinii in care apar evenimentele
        - Exemplul “make” : ceea ce conteaza este daca output.c este mai vechi sau mai nou decat output.o nu timpii lor absoluti de creare
-



---

# Ceasurile logice & ceasurile fizice

- Pentru numeroare aplicatii:
  - Este suficient ca toate masinile sa fie de acord asupra aceluiasi timp
  - Nu este esential ca acest timp sa fie in acord cu timpul real

Ex. make – este adecvat ca toate masinile sa fie de acord ca este 10:00 desi in realitate este 10:02.

Inteles: ceea ce conteaza este *consistenta internă a ceasurilor*, nu cat sunt de apropiate de timpul real.

Pentru acesti algoritmi conventia este aceea de a fi numite ***ceasuri logice***.
- Contra-exemplu:

Cand sunt constrangeri aditionale ca ceasurile

  - Nu numai sa fie aceleasi,
  - Dar sa nu devieze de la timpul real mai mult de o anumita cantitate

ceasurile sunt numite ***ceasuri fizice***.

# Alg. Lamport: sincronizarea ceasurilor logice

Lamport a definit relatia ***intamplat-inainte***.

- Expresia  $a \rightarrow b$ 
  - Se citeste " $a$  se intampla inainte de  $b$ "
  - Inseamna ca toate procesele sunt de acord ca prima data apare evenimentul  $a$ , si apoi apare evenimentul  $b$ .
- Relatia intamplat-inainte poate fi observata direct in situatiile urmatoare:
  1. Daca  $a$  si  $b$  sunt evenimente in acelasi proces, daca  $a$  apare inainte de  $b$  atunci  $a \rightarrow b$  este adevarat.
  2. Daca  $a$  este un eveniment de expedierea a unui mesaj de la un proces, si  $b$  este evenimentul de receptionare a mesajului la alt proces, atunci  $a \rightarrow b$  este adevarat.
    - inteles: un mesaj poate fi receptionat inainte de expediere sau in acelasi timp cu expedierea, deoarece necesita un timp finit pentru a fi ajunge

---

# Proprietatile relatiei intamplat-inainte

- Tranzitiv

- daca  $a \rightarrow b$  si  $b \rightarrow c$ , atunci  $a \rightarrow c$ .

- Daca doua evenimente,  $x$  si  $y$ , intamplate in procese diferite care nu schimba mesaje (nici indirect prin a treia parte),

- Atunci  $x \rightarrow y$  nu este adevarat, nici  $y \rightarrow x$ .

- Evenimentele sunt numite concurente

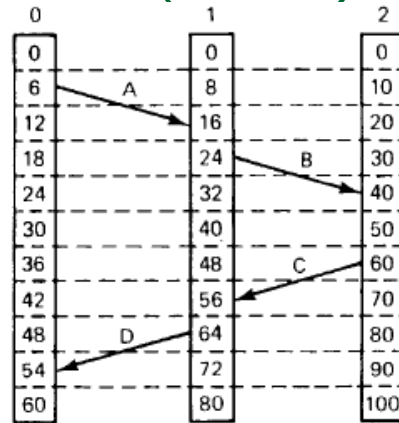
---

# Ceasul logic

- Pentru fiecare eveniment  $a$  se asigneaza o valoare in timp  $C(a)$  asupra caruia toate procesele sunt de acord.
- Proprietatea ca daca  $a \rightarrow b$ , atunci  $C(a) < C(b)$ .
  - Inteles:
    - daca  $a$  si  $b$  sunt doua evenimente din acelasi proces si  $a$  se intampla inaintea lui  $b$ , atunci  $C(a) < C(b)$ .
    - daca  $a$  este expedierea unui mesaj de catre un proces si  $b$  este receptionarea acelui mesaj la alt proces, atunci  $C(a)$  si  $C(b)$  trebuie asignate in asa fel incat toate procesele sunt de acord asupra valorilor  $C(a)$  si  $C(b)$  cu  $C(a) < C(b)$ .
- $C$  trebuie intotdeauna sa mearga inainte (creasca) si niciodata sa mearga inapoi (descreasca)
  - Corectiile in timp pot fi realizate prin adaugarea unei valori pozitive, niciodata prin scaderea uneiia

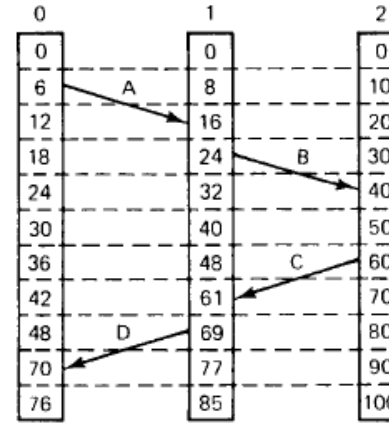
# Exemplu (1/2)

Ne-sinc



(a)

Sinc



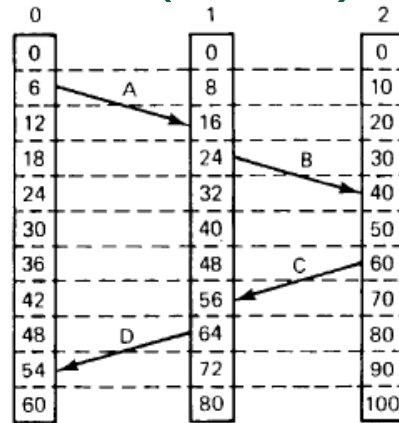
(b)

Stanga:

- 3 procese care ruleaza in masini diferite, fiecare cu ceasul propriu cu viteza proprie
- Cand ceasul a ticait de 6 ori in procesul 0, a ticait de 8 ori in procesul 1 si de 10 ori in procesul 2.
- Fiecare ceas ruleaza cu o rata constanta, dar ratele sunt diferite datoriat diferentelor in cristale
- Timpul 6: procesul 0 expediază mesajul A la procesul 1.
- Ceasul de la procesul 1 citește 16 cand acesta ajunge.
- Daca mesajul poarta si timpul de start, 6, procesul 1 concludă ca au fost necesare 10 ticuri pentru a ajunge mesajul la proces.
- Conform acestui fapt, mesajul B de la 1 la 2 poate necesita 16 ticuri, o valoare plauzibila.
- Mesajul C de la 2 la 1 pleaca de la 60 si ajunge la 56. Imposibil!
- Mesajul D de la 1 pleaca de la 64 si ajunge la 54. Imposibil!

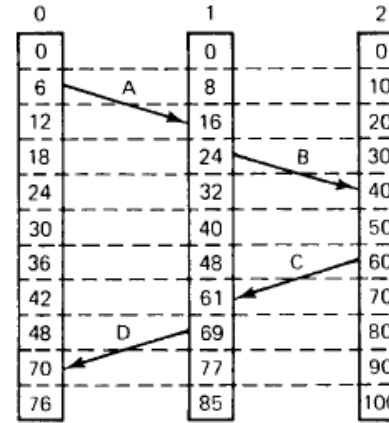
# Exemplu (2/2)

Ne-sinc



(a)

Sinc



(b)

Dreapta – Soluția Lamport:

- Urmărește relația întâmplat anterior.
- Deoarece C a plecat la 60, trebuie să ajungă la 61 sau mai târziu.
- De aceea, fiecare mesaj poartă și timpul de expediere, conform cu ceasul expeditorului
- Când un mesaj ajunge și ceasul receptorului arată o valoare mai timpurie decât timpul expedierii mesajului, destinatarul mută ceasul propriu la o valoare mai mare cu unu decât timpul de expediere
- Astfel mesajul C ajunge la 61. Similar, D ajunge la 70.

## Ajustari pentru a indeplini cerintele pt. timp global

- Intre oricare doua eveimente ceasul trebuie sa ticaie cel putin odata
  - Doua evenimente nu pot sa apara in acelasi timp
    - Abordare: ataseaza numarul de proces la care evenimentul a aparut la la timpul eveimentului, separat de punctul zecimal
      - ex. Evenimentele se intampla in procesele 1 si 2, ambele la timpul 40, primele devine 40,1 iar ultimul devine 40,2.
- > Asigneaza timpul la toate evenimentele dintr-un sistem distribuit este subiect al urmatoarelor conditii:
1. Daca  $a$  se intampla inainte de  $b$  in acelasi proces,  $C(a) < C(b)$ .
  2. Daca  $a$  si  $b$  reprezinta expedierea si receptionarea unui mesaj,  $C(a) < C(b)$ .
  3. Pentru toate evenimentele distincte  $a$  si  $b$ ,  $C(a)$  nu este egal cu  $C(b)$ .
- > O ordonare totala a tuturor evenimentelor in sistem.

---

# Necesitatea ceasurilor fizice

- Alg. lui Lamport pentru sincronizarea ceasurilor logice
    - Oferă o ordonare neambigua a evenimentelor,
    - Valorile în timp asignate evenimentelor nu sunt necesar apropiate de timpii reali la care apar
  - În anumite sisteme precum sistemele în timp real, ceasul real este important!
- > Pentru aceste sisteme ceasuri fizice externe sunt cerute
- Din rațiuni de eficiență și redundanță, ceasuri fizice multiple sunt în general dezirabile, ceea ce conduce la două probleme :
    1. Cum sunt sincronizate ceasurile cu cele reale, și
    2. Cum sunt sincronizate ceasurile între ele?[Următoarea săptămână!]
-



---

## Cum este de fapt masurat timpul? 1. Secunda solara

- Pana in secolul 17, timpul a fost masurat astronomic
  - Evenimentul la care soarele atinge cel mai ridicat punct pe cer este numit tranzit al soarelului – evenimentul apare la pranz in fiecare zi.
  - Intervalul intre doua tranzite consecutive ale soarelui este numita zi solara.
  - Cum se considera 24 de ore intro zi, fiecare continand 3600 secunde, *secunda solara* este definita ca fiind a 86400 parte a unei zile solare.
- Anii 1940: perioada de rotatie a Pamantului nu este constanta!
  - Pamantul incetinesc datorita frictiunii datorate atmosferei si mareelor
  - Geologii sustin ca acum 300 de milioane de ani erau 400 zile pe an in sens matematic
  - Lungimea anului, timpul pentru o rotatie in jurul soarelui se considera neschimbata; ziua a devenit mai lunga
  - In plus pe langa acest trend pe termen lung, variatii scurte in lungimea zilei apar si datorita turbulentelor de adancime in miezul Pamantului
  - Astronomii au calculat lungimea zilei masurand un numar mare de zile si si considerarea mediei inainte de impartirea cu 86400 – rezultand cantitatea numita *secunda solara medie*.

---

## Cum este masurat timpul? 2. TAI

- Inventia ceasului atomic in 1948 => masurarea timpului cu o precizie mai mare,
  - independent fenomenele la care Pamantul este supus,
  - contorizeaza tranzitiile atomului de cesiu 133.
- Fizicienii: definesc secunda ca fiind timpul necesar atomului de cesiu 133 sa faca exact 1770 tranzitii.
  - Alegerea a fost realizata astfel incat secunda atomica a fost egala cu secunda solara medie in anul introducerii sale.
  - Mai mult de 50 laboratoare din intreaga lume au ceasuri cu cesium 133.
  - Periodic, fiecare laborator spune Biroului International al orei (BIH) din Paris cat timp a ticait ceasul lui.
  - BIH face media acestor valori si produce *Timpul Atomic International*, abreviat TAI.
  - TAI este numarul mediu de ticuri ale ceasurilor bazate pe cesiu 133 din noaptea lui 1 Ian 1958 (inceputul timpului) impartit la 1770.

---

# Cum este masurat timpul? Problema TAI

- TAI este stabil si disponibil oricaruia doreste sa cumpere un ceas cu cesiu,
- DAR problema seroasa este aceea ca: 86 400 secunde TAI sunt cu 3 msec mai putin decat ziua solara in medie deoarece ziua solara medie devine din ce in ce mai lunga
- Abordare?
  - Solutia Papei Gregor al XIIIlea: in 1582 a decretat ca 10 zile sa fie omise din calendar.
    - Acest eveniment a cauzat rascoale pe strafa deoarece latifundierii au cerut renta pe toata luna si la fel si camatarii, pe cand angajatii au refuzat sa plateasca pentru cele 10 zile pe care nu le-au lucrat
    - Tarile protestante, din principiu, au refuzat sa aiba de a face cu decretele papale si nu au acceptat calendarul Gregorian pentru 170 de ani.

---

## Cum este masurat timpul? 3. UTC

- BIH rezolva problema prin introducerea *secundelor de saritura* cand discrepanta intre TAI si timpul solar creste peste 800 msec.
- Aceasta corectie conduce la un sistem de timp bazat pe secundele TAI dar care sta in faza cu miscarea aparaneata a soarelui
- Este numit *Timp Coordonat Universal*, abreviat UTC.
- UTC este baza timpului modern civil.
- A inlocuit standardul vechi, Timpul mediu Greenwich (GMT), care este timpul astronomic

---

# Cunoasterea UTC

- Majoritatea companiilor de curent electric isi bazeaza ceasurile lor de 60-Hz sau 50-Hz pe UTC
- Cand BIH anunta o secunda de salt, companiile ridica frecventa la 61 Hz sau 51 Hz pentru 60 sau 50 sec, pentru a avansa toate ceasurile din aria lor de distributie
- 1 sec este un interval semnificativ pentru un calculator si SO trebuie sa tina timpul cu acuratete pentru o perioada de ani astfel incat trebuie sa considere software special pentru a tine seama de secunde de salt cand acestea sunt anuntate
- UTC este difuzat prin radio, sateliti etccu o acuratete de 0.5msec
  - Trebuie cunoscuta cu acuratete pozitia relativa a expeditorului si destinatarului, pentru a compensa pentru intarzierea propagarii semnalului