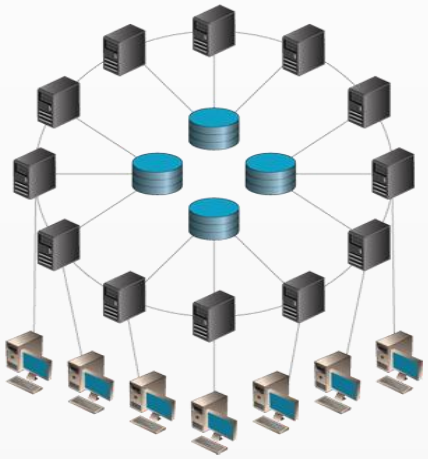
Sincronização de Relógios e Exclusão Mútua

SMD0050 - SISTEMAS DISTRIBUÍDOS



Slides são baseados nos slides do Couloris e Tanenbaum

Sincronização de Relógios

- Relógios são essenciais no uso de computação, seja para medir o tempo ou identificar seqüências de eventos diversos.
  - Em sistemas centralizados, o tempo não é ambíguo, sendo gerenciado em apenas 1 máquina e obtido por chamada ao núcleo;
  - Em sistemas distribuídos, obter um horário comum a vários computadores **não é trivial**.

Andrew S. Tanenbaum
Maarten Van Steen

Quais seriam as dificuldades de obter um mesmo horário em um sistema distribuído?

Relógios físicos – Medida do tempo

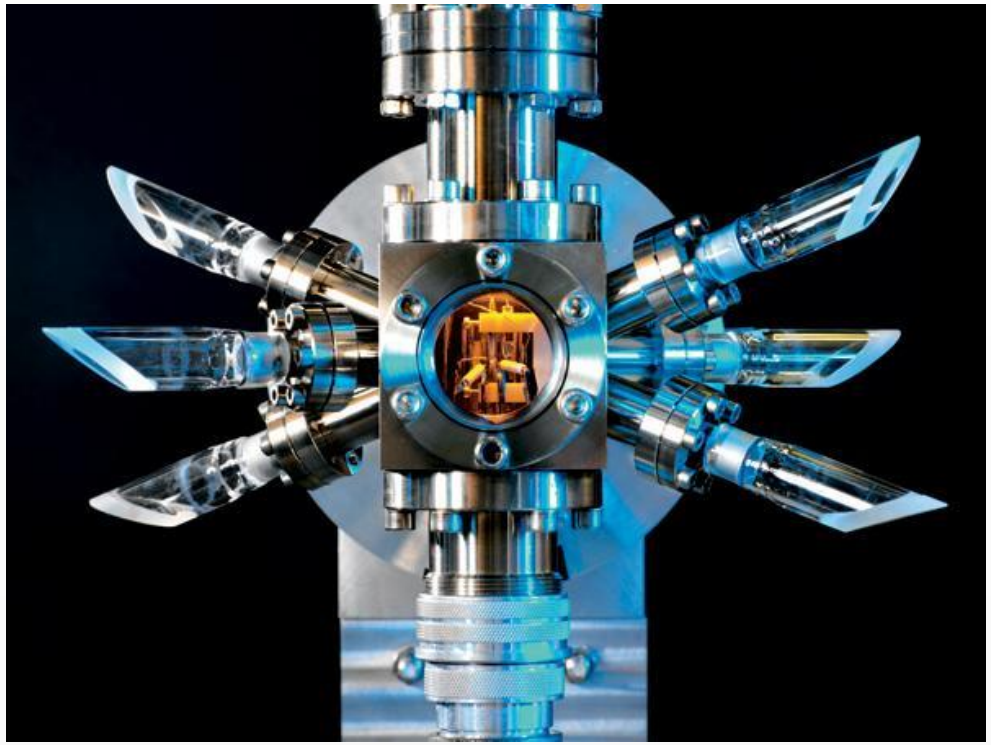
- Até invenção dos relógios mecânicos (sec. XVII):
  - Tempo medido com auxílio dos astros;
    - Sol nasce no horizonte a leste;
    - Alcança uma altura máxima no céu, ao meio dia (trânsito solar);
    - Sol se põe no horizonte a oeste;
  - Intervalo entre 2 trânsitos solares consecutivos do Sol é um dia solar;
  - Um segundo solar é exatamente 1/86.400 de um dia solar.
  - Cada hora possui 3600 segundos;

Relógios Físicos – Medida de tempo

- Em meados de 1940, estabelecido que a rotação da Terra não é constante
  - Devido à desaceleração gradativa resultante de marés e atrito com atmosfera;
  - Há 300 milhões de anos o ano tinha 400 dias!
    - A Terra gira mais devagar, mas não alterou sua órbita.
    - Logo, o tamanho do ano aparenta ser o mesmo, mas os dias ficaram mais longos!
- Necessário definir nova medida para o tempo:
  - Calculado o tempo de vários dias, obtendo uma média do dia.
  - Ao dividi-la por 86.400, obtém-se o segundo solar médio.

Relógios Físicos – Medida de tempo

- Em 1948 foi inventado o relógio atômico:
  - Calcula o tempo através de contagens de transições do césio 133 (1 segundo solar médio = 9.192.631.770 transições).
- Tal fato tornou possível:
  - Medir o tempo com maior precisão;
  - Medi-lo independentemente das condições do globo terrestre e da atmosfera.
  - Cálculo da hora atômica internacional (TAI) pelo BIH.



Relógio do NPL do Reino Unido*

* https://hypescience.com/este-belo-relogio-atomico-e-governado-por-um-unico-atomo-de-estroncio/

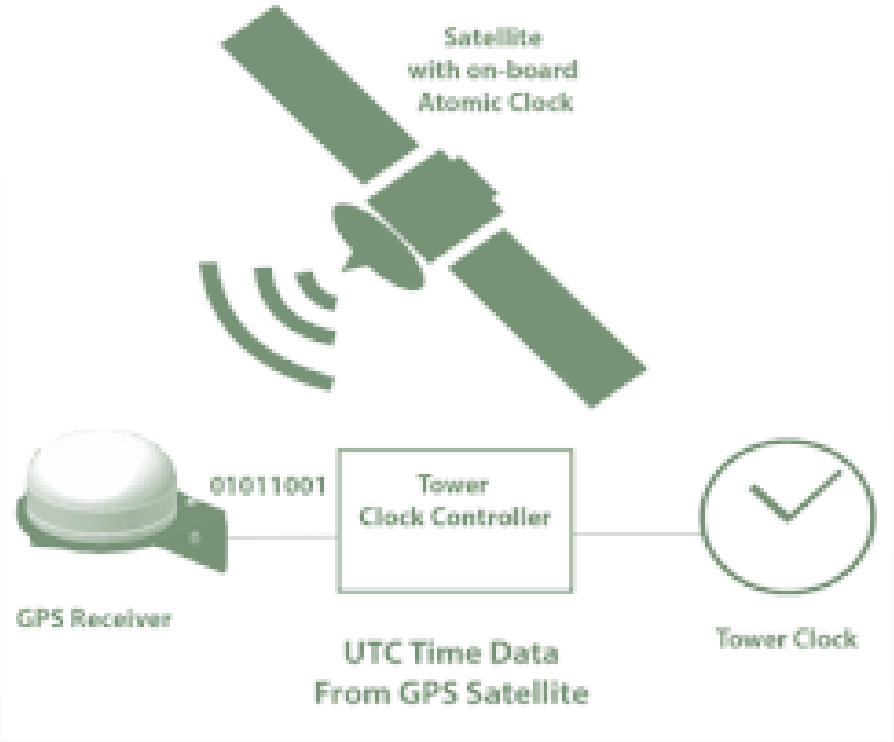
Relógios físicos – Medida do tempo

- Problema:
  - 86.400 segundos TAI equivalem a aproximadamente 3ms a menos que um dia solar médio.
- Solução:
  - Segundos extras a cada 800ms acumulados

Figura 6.3 Segundos TAI têm comprimento constante, diferentes dos segundos solares. Os segundos extras são introduzidos quando necessário para se manterem em fase com o sol.

Relógios Físicos e GPS

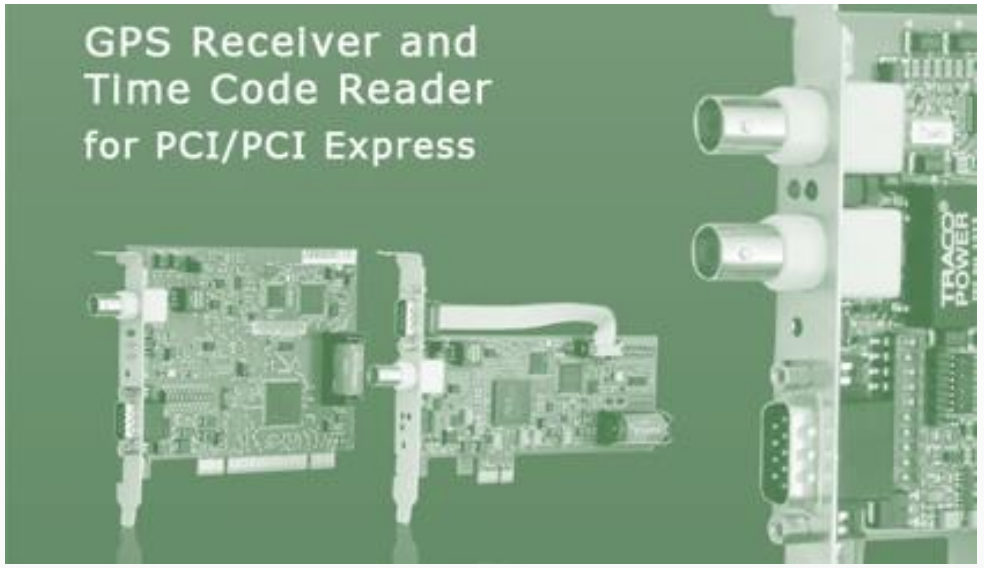
- Com base no TAI com correções de segundos foi estabelecido o sistema UTC (Universal Coordinated Time), que é a base de toda a moderna medição de tempo.
  - O Nist fornece UTC através de rádios de ondas curtas (WWV)
  - Além disso vários satélites fornecem UTC



O GPS faz triangulação usando satélites de modo a calcular as diferenças de tempo entre o UTC de cada satélite usado, calculando, assim, a localização geográfica do ponto

Uma solução para sincronismo de máquinas distribuídos

- Se uma das máquinas possui receptor WWV ou GPS
- Manter todas as outras máquinas sincronizadas com ela.

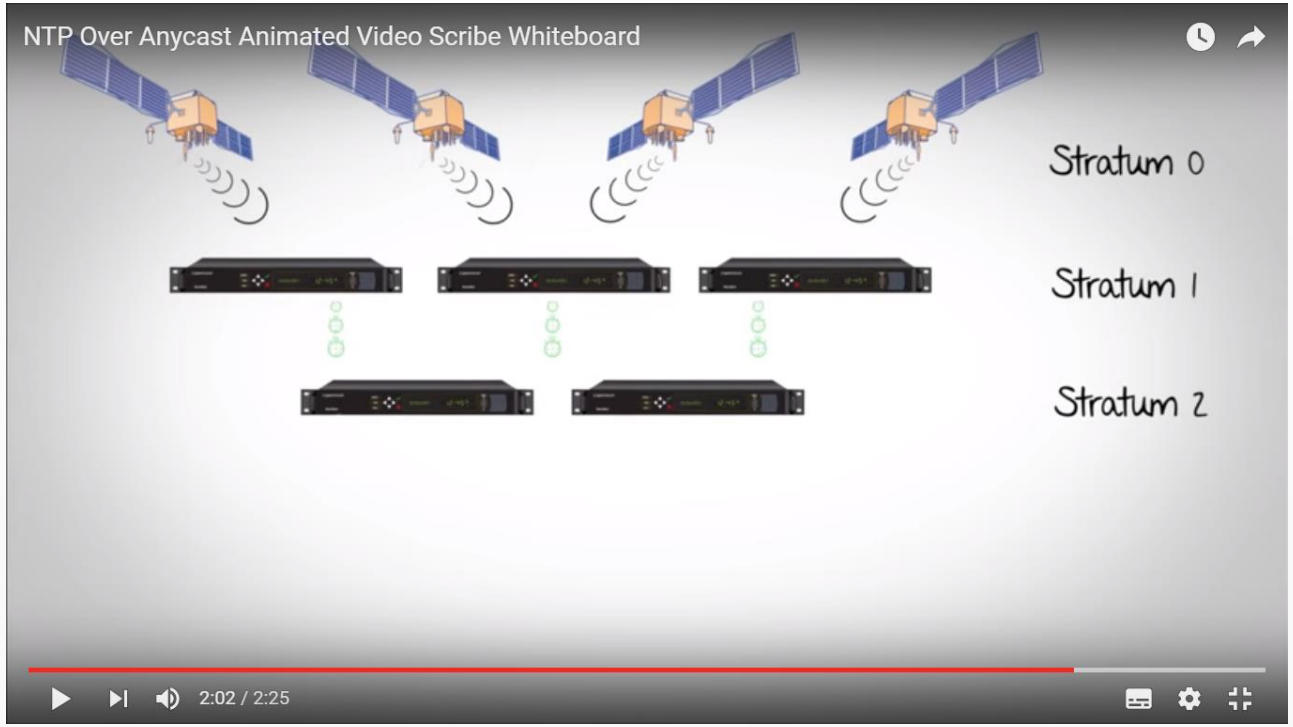


Algoritmos de sincronização de relógios

- Se nenhuma possui receptor WWV ou GPS
  - cada uma cuida de seu próprio horário;
  - o problema passa a ser manter o horário de todas máquinas o mais próximo possível.
- Foram propostos vários algoritmos, todos seguindo as mesmas idéias básicas
- Como levar em contar o delay da Rede?

Protocolo de tempo de Rede - NTP

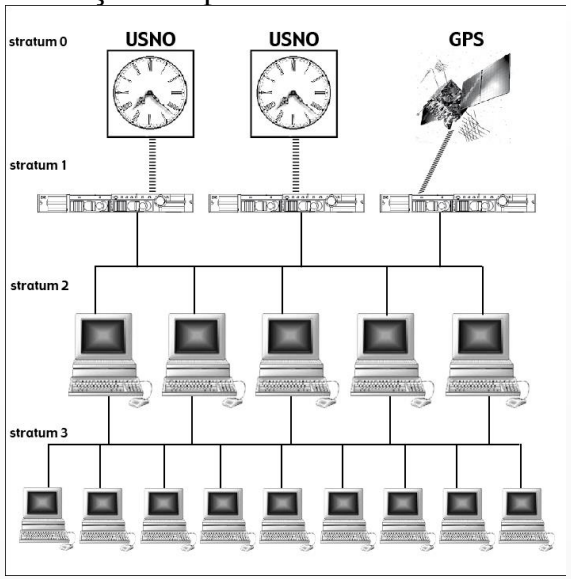
- NTP over anycast



https://www.youtube.com/watch?v=GaYJ3CXXZE4

Como funciona o NTP?

A Figura abaixo ilustra uma hierarquia de funcionamento do NTP (Network Time Protocol) , no qual o stratum 0 é composto de relógios atómicos mantidos pela Marinha americana (USNO) ou por relógios existentes no sistema de satélites que dão suporte ao GPS. De posse dessas informações responda:



a) A razão de ser do NTP está relacionada às dificuldades para se obter um consenso sobre o tempo físico em um sistema distribuído mesmo que todos os computadores tenham seus relógios sincronizados manualmente em um instante t. Explique quais são essas dificuldades e exemplifique um cenário que demonstra essas dificuldades (empregue o conceito de tempo relativo).
b) Explique como funciona o protocolo de tempo de rede (NTP), evidenciando o problema desta solução e como amenizar o mesmo. Use a figura como guia da explicação.

Protocolo de tempo de Rede - NTP

- Proposto por Cristian(1989), baseia-se em clientes consultarem um servidor de tempo.
- Funcionamento:
  1. Cada máquina envia uma mensagem para o servidor de tempo (máquina com receptor WWV ou relógio de precisão), perguntando pelo tempo corrente
  2. Servidor de tempo responde o mais rápido possível, com uma mensagem contendo o tempo corrente C_{UTC}
  3. Quando o transmissor obtém uma resposta, ajusta seu clock.
- Problemas:
  - O tempo não pode retroceder;
  - Há atraso no envio das mensagens.

Protocolo de tempo de Rede - NTP

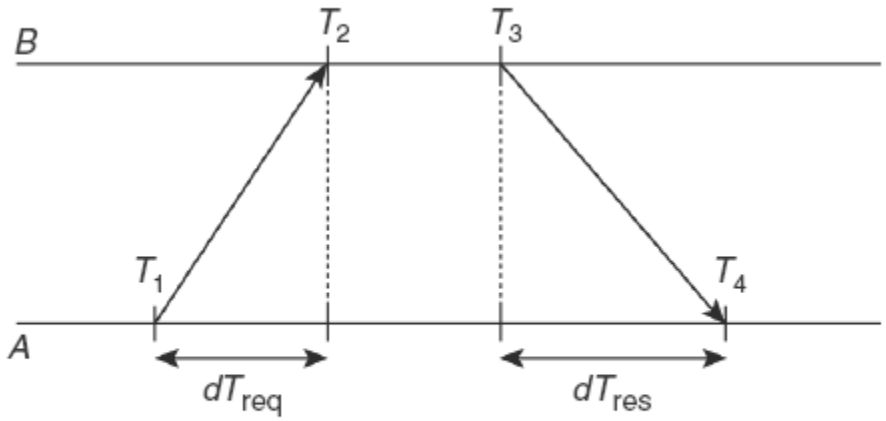


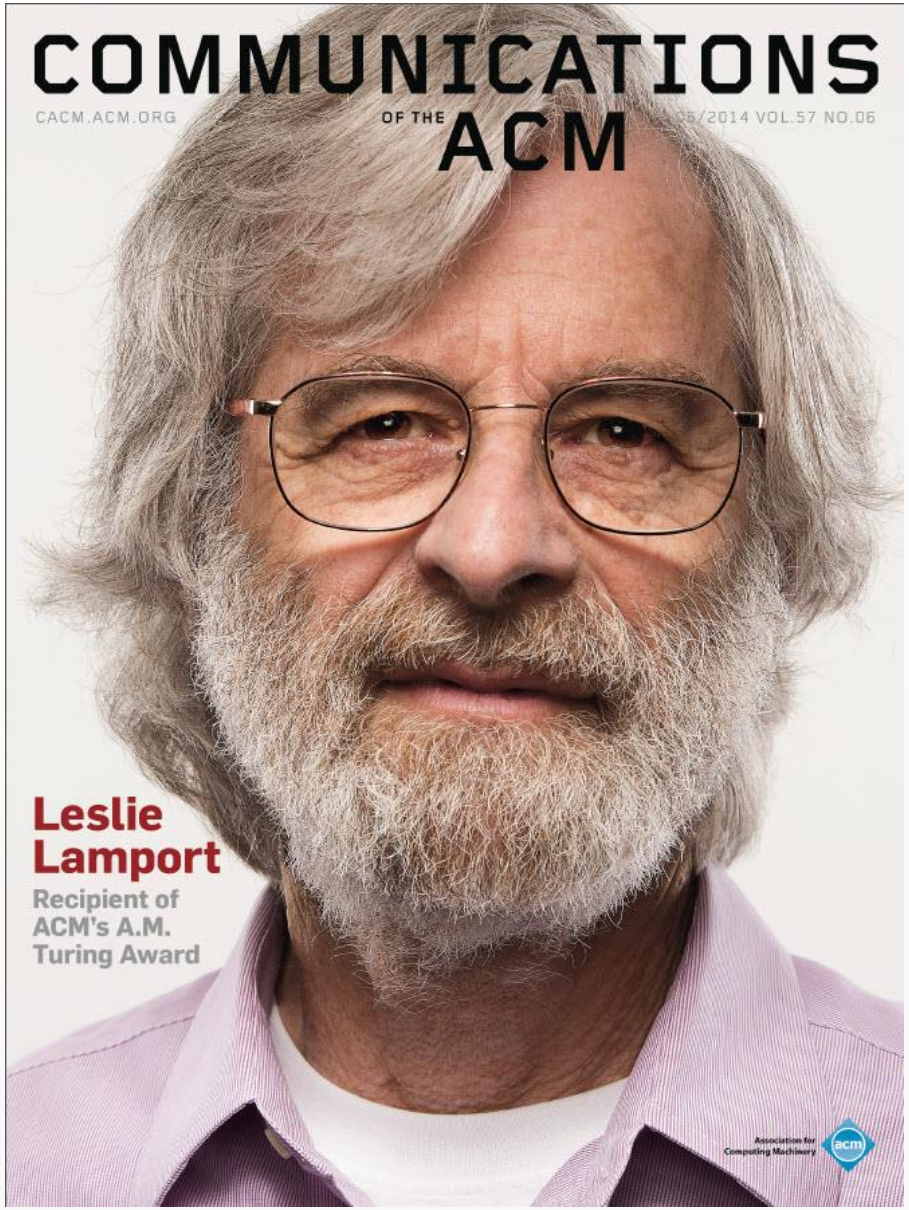
Figura 6.6 Obtenção da hora corrente por meio de um servidor de tempo.

- $\theta=[(T2-T1)+(T3-T4)]/2$
  - Será o tempo de erro entre os relógios
- $\delta=[(T2-T1)-(T4-T3)]/2$
  - Será o atraso estimado do envio

Relógios Lógicos

- Até o momento foi considerada a sincronização de relógios como naturalmente relacionada com a hora real.
  - Lamport mostrou que, embora a sincronização de relógios seja possível, não precisa ser absoluta.
- Soluciona problemas relativos a ordem de eventos independentemente da hora real

Não é necessário sincronizar processos que não interagem entre si



Relógios Lógicos
Problemas na ordenação de eventos

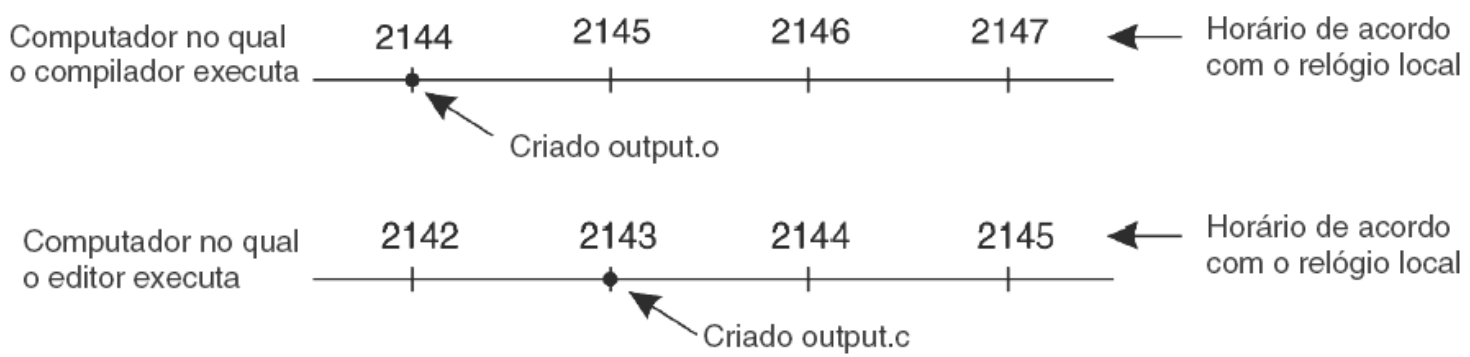


Figura 6.1 *Quando cada máquina tem seu próprio relógio, um evento que ocorreu após outro evento pode, ainda assim, receber um horário anterior.*

Relógios Lógicos de Lamport

- Definir a relação acontece antes
- Se a e b são eventos do mesmo processo:
  - Se a acontece antes de b: $a \rightarrow b$.
  - Sendo a é o envio da mensagem e b o recebimento da mensagem: $a \rightarrow b$.
  - Se $a \rightarrow b$ e $b \rightarrow c$, então $a \rightarrow c$ (propriedade transitiva).

Se x e y acontecem em processos diferentes que não trocam mensagens, então tanto $x \rightarrow y$ quanto $y \rightarrow x$ são falsas! Esses processos são ditos concorrentes.

- Tempos são medidos em função: se $a \rightarrow b$, $C(a) < C(b)$

Relógios Lógicos de Lamport

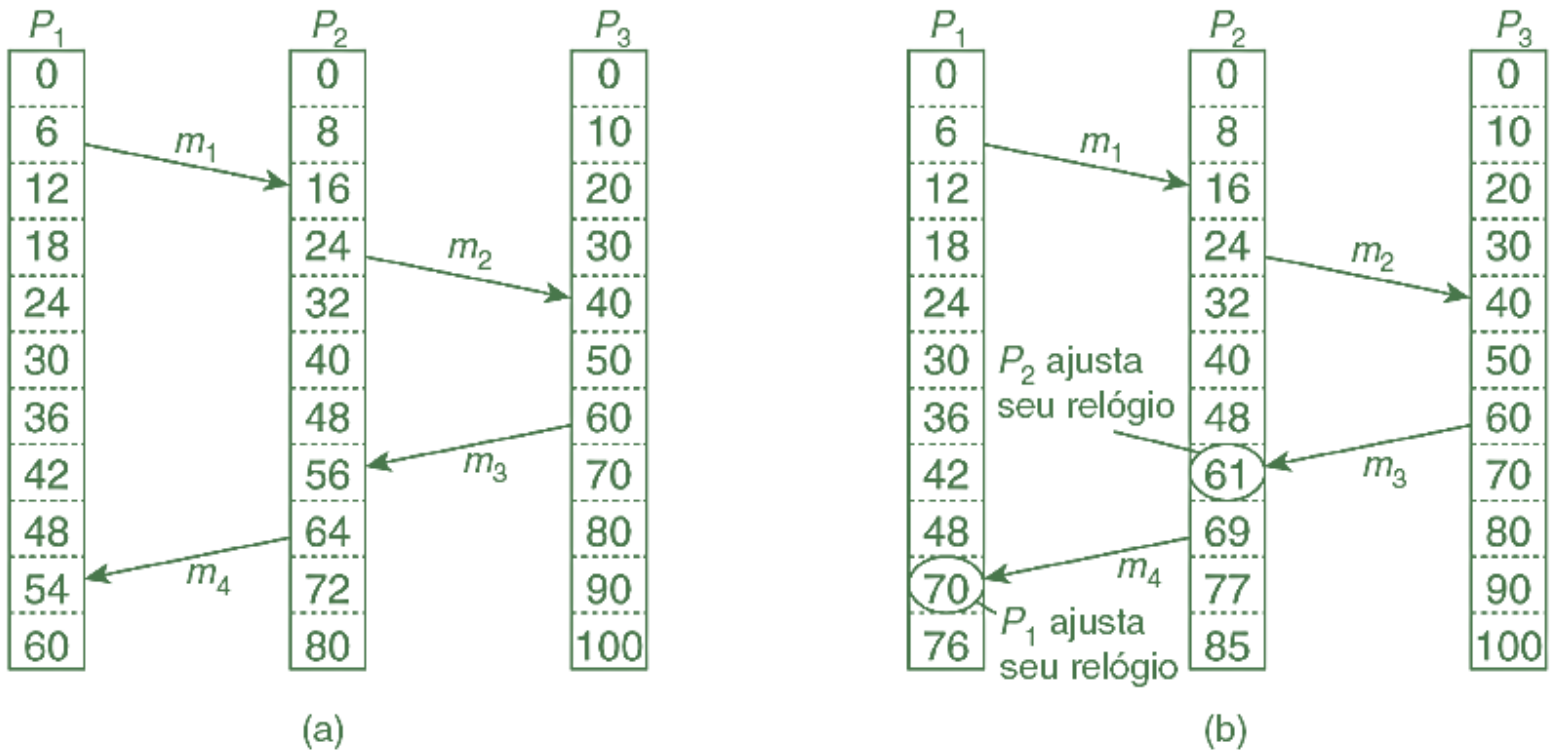


Figura 6.9 (a) Três processos, cada um com seu próprio relógio. Os relógios funcionam a taxas diferentes. (b) O algoritmo de Lamport corrige os relógios.

Arquitetura - Relógios Lógicos de Lamport

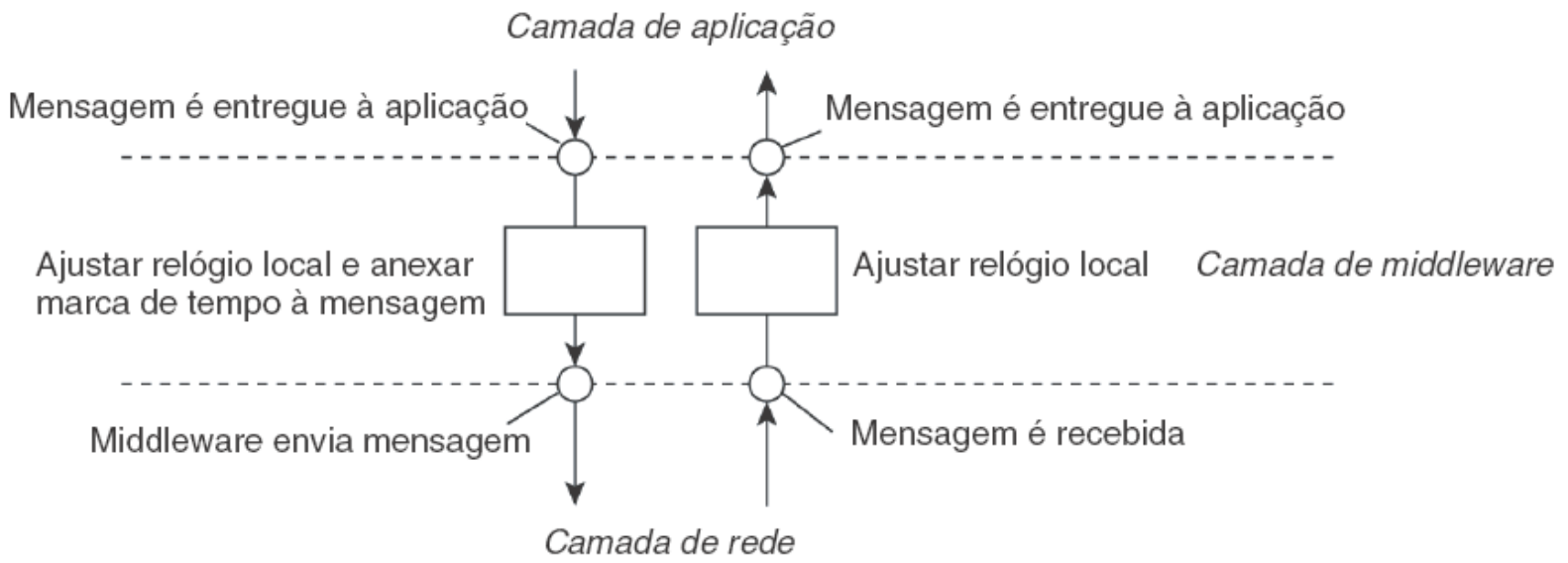


Figura 6.10 Posicionamento de relógios lógicos de Lamport em sistemas distribuídos.

Andrew S. Tanenbaum
Maarten Van Steen

Relógios Lógicos de Lamport

- Ordenação total de eventos: Uma operação multicast pela qual todas as mensagens são entregues na mesma ordem a cada receptor.
  - Cada mensagem será enviada em multicast e sempre transportará a marca de tempo (lógico) de seu remetente;
  - Mensagens são ordenadas em fila de cache local pela marca lógica de tempo;
- Quando uma mensagem é recebida, a mesma é adicionada a seu cache local, e uma mensagem de reconhecimento é enviada em multicast;
  - Mensagens só podem ser entregues à aplicação após reconhecimento de todos processos, apenas quando forem a primeira mensagem da fila;

Uso em Multicast Totalmente Ordenado

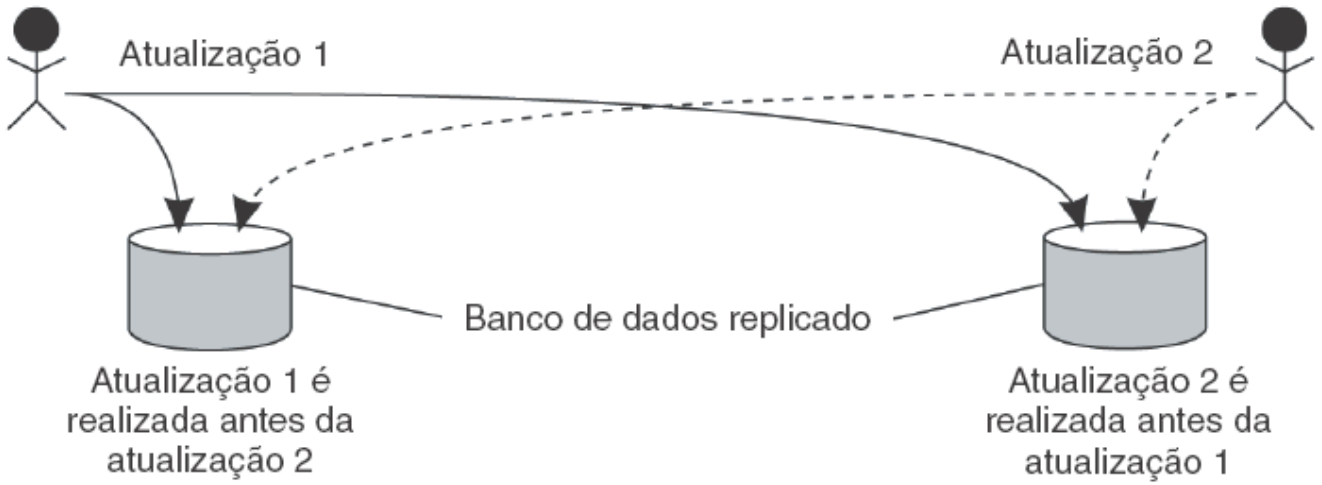


Figura 6.11 Atualização de banco de dados replicado que o deixa em estado inconsistente.

Andrew S. Tanenbaum
Maarten Van Steen

Algoritmos de Exclusão Mútua

Exclusão Mútua

- Como garantir que o acesso concorrente de recursos não gere situações de inconsistência de dados?
- A *Exclusão Mútua*, que, em S.D. pode ser dividida em duas categorias:
  - *Baseadas em fichas*
    - Evita inanição (Starvation)
    - Fácil evitar deadlocks
  - *Baseadas em permissão*
    - Processo que quer recursos deve antes pedir permissão aos demais

Exclusão Mútua- Algoritmo centralizado

- Simula o que é feito em um sistema monoprocessador
  - Um processo é eleito como coordenador
  - Sempre que um processo quiser acessar determinado recurso, é necessário pedir permissão
    - Mensagem enviada ao coordenador
  - O coordenador permite acesso ao recurso através de uma mensagem de concessão
    - Desde que nenhum outro processo esteja acessando o recurso neste momento

Algoritmo centralizado

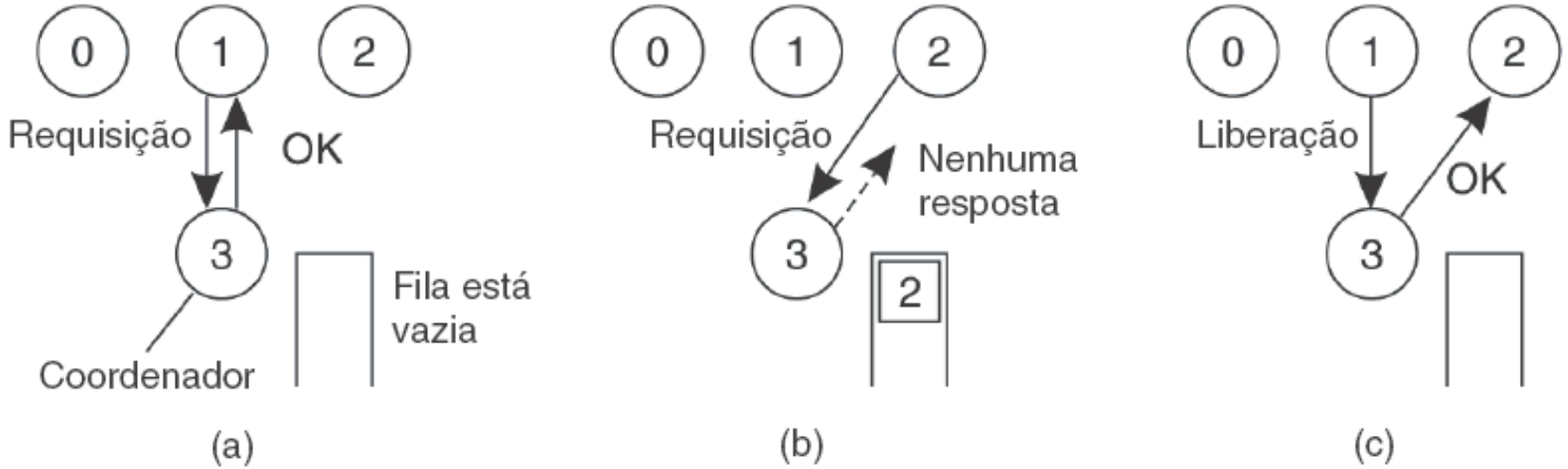


Figura 6.14 *(a) O processo 1 solicita ao coordenador permissão para acessar um recurso compartilhado. A permissão é concedida. (b) Depois, o processo 2 solicita permissão para acessar o mesmo recurso. O coordenador não responde. (c) Quando o processo 1 libera o recurso, informa ao coordenador, que então responde a 2.*

Vantagens e Desvantagens

- Prós
  - Simples, fácil de entender e de implementar;
  - É justo (segue o FCFS - First come First served);
  - Como no FCFS, não há inanição (starvation)
- Contras
  - Como todo sistema centralizado: um ponto de erro e um ponto de gargalo
  - Processos não conseguem distinguir coordenador inativo de permissão negada

Algoritmo Totalmente Descentralizado

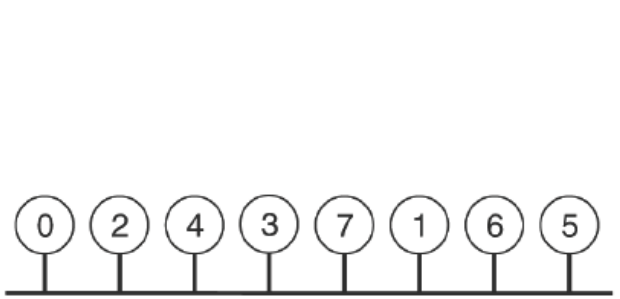
- Usar um algoritmo de votação em um sistema baseado em DHT – Distributed Hashed Table:
  - Cada recurso é replicado n vezes;
  - Cada réplica possui um coordenador;
  - Processo requisitante precisa receber voto majoritário de coordenadores: $m > n/2$
  - Caso a permissão seja negada (processo obtém menos que m votos), o mesmo desistirá do recurso por um período de tempo aleatório, e fará a tentativa novamente mais tarde.

Algoritmo descentralizado

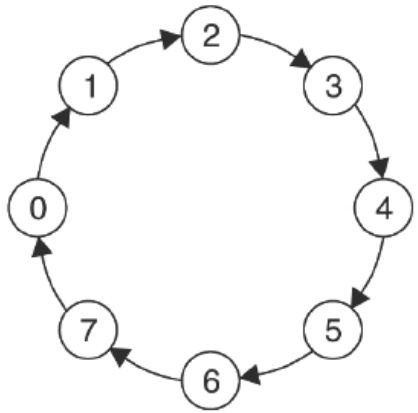
- Prós
  - Torna a solução centralizada original menos vulnerável a falhas de um único coordenador;
  - Possibilita uso de réplicas de recursos;
- Contras
  - Não protege contra starvation;
  - Se muitos nós querem acessar o mesmo recurso, nenhum nó conseguirá votos suficientes, e os recursos deixarão de ser usados;
  - Há uma probabilidade positiva (embora muito baixa) de permitir a 2 nós acesso ao mesmo recurso, ao mesmo tempo (ou seja, não garantir a exclusão mútua).

Exclusão Mútua - Algoritmo Token Ring

- Baseado nas redes *token ring* estudadas em rede, mas não é necessariamente uma rede física: pode ser uma rede *lógica*



(a)



(b)

Figura 6.16 (a) Grupo de processos não ordenados em uma rede. (b) Um anel lógico é construído em software.

Andrew S. Tanenbaum
Maarten Van Steen

Exclusão Mútua - Algoritmo distribuído

- Para muitos, um algoritmo correto segundo as leis da probabilidade não é bom o bastante (incluindo o professor).
- Para resolver esse problema, pesquisadores procuraram algoritmos distribuídos determinísticos de exclusão mútua.
  - Lamport apresentou o primeiro em 1978;
  - Ricart e Agrawala o tornaram mais eficiente (1981).

Exclusão Mútua - Algoritmo distribuído

- Requer ordenação total de todos os eventos no sistema;
  - Para isso será usado... Algoritmo de Lamport!
- Funcionamento:
  - Quando processo deseja acessar um recurso compartilhado, monta uma mensagem que contém o nome do recurso, seu número de processo e a hora corrente (lógica).
  - Envia mensagem para todos processos, inclusive ele mesmo (similar a broadcast ou multicast);
  - Quando um processo recebe uma mensagem de requisição de outro, executa uma ação de acordo com seu próprio estado em relação ao recurso:

Exclusão Mútua - Algoritmo distribuído

- Funcionamento (continuação):
  - Se o receptor não estiver acessando o recurso nem quer acessá-lo, devolve "OK" ao remetente
  - Se já tem acesso ao recurso, não responde e coloca requisição em uma fila;
  - Se receptor também quer acessar o recurso, mas ainda não possui a permissão, compara a marca de tempo da mensagem que chegou com a marca de tempo da mensagem que enviou a todos. A mais baixa vence.
- O processo remetente aguarda recebimento de todas as respostas
  - Quando houver permissão de todos, processo acessa o recurso;
  - Processo libera o recurso enviando um "OK" a todos os processos

Exclusão Mútua - Algoritmo distribuído

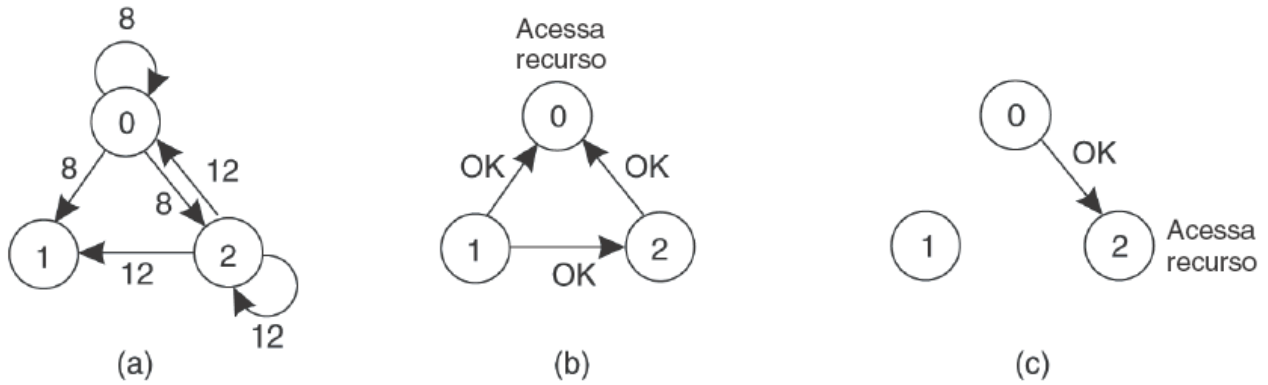


Figura 6.15 (a) Dois processos querem acessar um recurso compartilhado no mesmo momento. (b) O processo 0 tem a marca de tempo mais baixa, portanto vence. (c) Quando o processo 0 conclui, também envia uma mensagem *OK*, portanto, agora, 2 pode seguir adiante.

Exclusão Mútua - Algoritmo distribuído

- Vantagens:
  - Exclusão mútua é garantida;
  - Não há deadlock;
  - Não há starvation;
  - Número de mensagens: 2(n-1) //n=número de nós
  - Não existe nenhum ponto de falha único

Exclusão Mútua - Algoritmo distribuído

- Desvantagens
  - Ponto de falha único foi substituído por n pontos de falha;
  - Deve usar comunicação multicast ou manter uma lista de associação ao grupo em cada processo, incluindo processos que entram no grupo ou caem (funciona melhor com poucos processos que se mantêm estáveis);
  - Trocou 1 gargalo por n gargalos;
  - Portanto é: Mais lento, mais complicado, mais caro e menos robusto que o original centralizado.

Exclusão Mútua - Comparação entre os quatro algoritmos

Algoritmo	Mensagens por entrada/saída	Atraso antes da entrada (em número de tempos de mensagens)	Problemas
Centralizado	3	2	Queda do coordenador
Descentralizado	$3mk$, $k = 1,2,\ldots$	$2m$	Inanição, baixa eficiência
Distribuído	$2(n-1)$	$2(n-1)$	Queda de qualquer processo
Token Ring	1 a ∞	0 a $n-1$	Ficha perdida; processo cai

Tabela 6.1 Comparação entre quatro algoritmos de exclusão mútua.

Tarefa de Casa

- Implementar um algoritmo centralizado de exclusão mútua
  - Existe um servidor no endereço X que contém informações do saldo de um usuário
  - Aplicações nos endereços X, Y, Z só podem modificar esse objeto se obtiverem permissão do coordenador
  - Usem Sockets ou MQTT na implementação