Sistemas Distribuídos

Sincronização

Referência

- “Sistemas operacionais modernos” Andrew S. TANENBAUM Prentice-Hall, 1995
  - Seção 11.1 pág. 316 - 325

Conteúdo

- Relógios lógicos
  - Algoritmo de Lamport
- Relógios físicos
- Algoritmos para sincronização de relógio
  - Algoritmo de Cristian
  - Algoritmo de Berkeley
  - Algoritmo baseado na média
  - Sistema com várias fontes externas

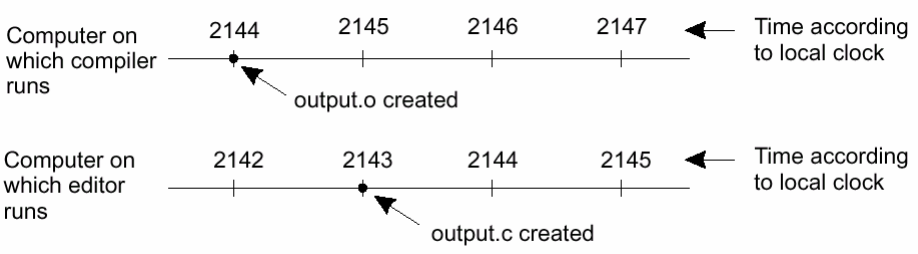
Sincronização através do relógio

- Sincronização:
  - sistemas distribuídos ≠ sistemas centralizados
- Sistemas distribuídos utilizam algoritmos distribuídos não é conveniente coletar todas as informações sobre a situação corrente para então tomar as decisões

Sincronização através do relógio

- Propriedades dos algoritmos distribuídos:
  1. Informações espalhadas em diversas máquinas
  2. Processos decidem baseados em informações locais
  3. Deve ser evitado um único ponto de falha
  4. Não existe relógio global
- Ex.: um único processo gerenciando acesso a E/S
- Ex.: programa *make*

Sincronização através do relógio



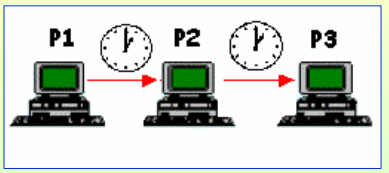
Sincronização através do relógio

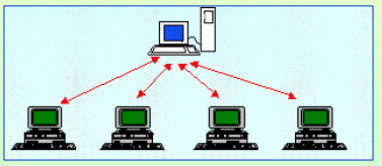
- **Por que é necessário a sincronização de clocks?**

A ordem cronológica entre 2 eventos é importante.

- **Clock Lógico:** consistência interna entre os clocks

- **Clock Físico:** consistência interna entre os clocks e com o tempo real





- Qual a solução para a sincronização de clocks?

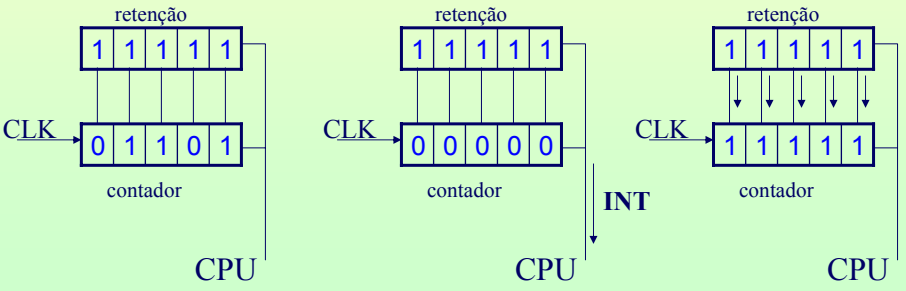
Algoritmos de Lamport, Cristian, Berkeley e Daemon Time-Manager

Relógios lógicos

- Temporizador: dispositivo baseado em cristal de quartzo que oscila em uma freqüência bem definida. Dois registradores são associados ao temporizador:
  - contador: quando chega a zero gera interrupção
  - retenção: valor carregado no contador
- Escorregamento de clock: diferenças entre as freqüências dos cristais que fazem com que os clocks percam o sincronismo.

Relógios lógicos

- **O que é CLOCK?**
  - Cristal de quartzo;
  - Registrador Contador;
  - Registrador de Retenção;



Relógios lógicos

- É possível sincronizar TODOS os relógios para obter um tempo padrão único?

- Lamport (1978):
  - se 2 processos não interagem não é necessário sincronismo
  - não é preciso que todos os processos estejam de acordo com o valor exato do tempo mas apenas sobre a ordem em que os eventos acontecem

Relógios lógicos

- Relógios lógicos: medem a passagem do tempo mas não precisa estar com o valor igual ao da hora oficial
- Relógios físicos: dispositivos que não podem diferir da hora oficial mais do que um determinado valor

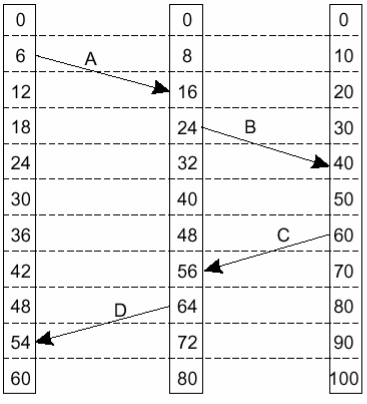
Algoritmo de Lamport

- Relação acontecimento-anterioridade: a $\rightarrow$ b (a acontece antes de b). Pode ocorrer em 2 situações:
  - Se a e b são eventos do mesmo processo, e se a ocorre ante de b, então a $\rightarrow$ b é verdadeira.
  - Se a é o evento de uma mensagem sendo enviada e b é o evento da mesma mensagem sendo recebida , então a $\rightarrow$ b é verdadeira.

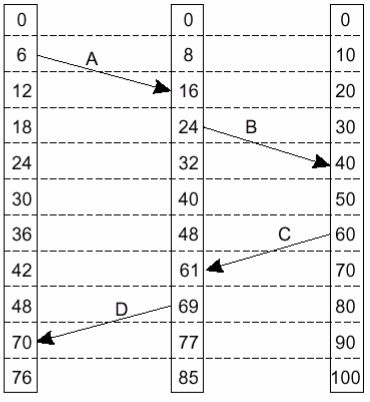
Algoritmo de Lamport

- Relação transitiva:
  se a $\rightarrow$ b e b $\rightarrow$ c então a $\rightarrow$ c
- Eventos concorrentes:
  - se x e y são eventos que acontecem em processos distintos que não trocam mensagens (nem mesmo indiretamente) então nada pode ser dito sobre x $\rightarrow$ y e y $\rightarrow$ x.
- Se a $\rightarrow$ b então C(a) < C(b)
- O tempo medido pelo clock é crescente. Para corrigi-lo deve-se somar.

Algoritmo de Lamport



(a)



(b)

Algoritmo de Lamport

- Com uma pequena modificação, cumpre os requisitos para a implementação do tempo global:
  - Entre 2 eventos, o clock precisa rodar pelo menos 1 vez (mesmo que os eventos aconteçam em rápida sucessão)
  - Como 2 eventos não podem nunca ocorrer exatamente no mesmo instante, deve-se juntar o número do processo no qual o evento ocorre aos bits de mais baixa ordem do tempo

Algoritmo de Lamport

- Podemos atribuir o tempo de todos os eventos sujeito às seguintes condições:
  1. Se a ocorre antes de b no mesmo processo, então $C(a) < C(b)$.
  2. Se a e b são envio e recepção de uma mensagem, então $C(a) < C(b)$.
  3. Para quaisquer eventos a e b, $C(a) \neq C(b)$.

Relógios físicos

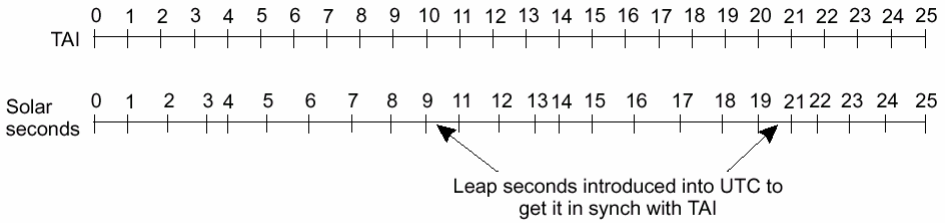
- Trânsito do sol: evento do Sol alcançando seu ponto de maior altura no céu
- Dia solar: intervalo entre 2 trânsitos consecutivos
- Segundo solar: dia solar/86.400

  86.400 = 24 * 3600
- Segundo solar médio: dia solar médio/86.400 (necessário pois o dia solar varia)

Relógios físicos

- Relógios atômicos de césio: tempo para que o césio 133 faça 9.192.631.770 transições
- 86.400 segundos TAI (Tempo Atômico Internacional) é, atualmente, 3ms menor que o dia solar
- Foram introduzidos os segundos bissextos (quando a diferença atinge 800 ms) → UTC (Tempo Universal Coordenado)
- O UTC está substituindo o GMT (Tempo Médio de Greenwich)

Relógios físicos



Relógios físicos

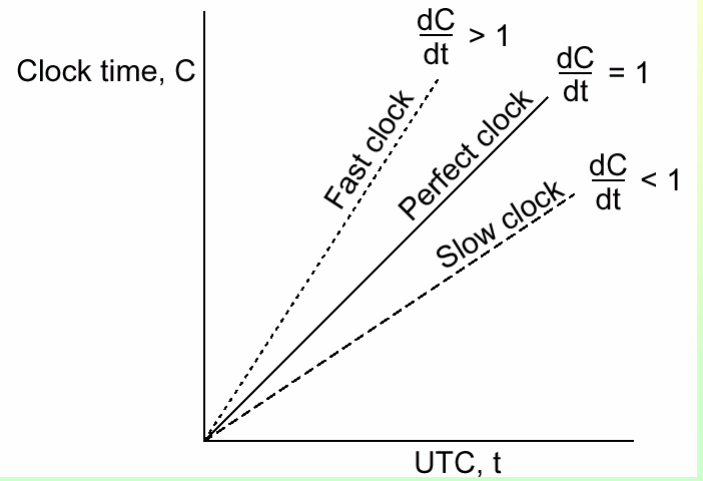
- Disponibilização do UTC:
  - Estação de rádio de ondas curtas prefixo WWV (NIST)
  - Satélite GEOS
  - Serviço na Internet: time.nist.gov

Algoritmo para sincronização de relógio

- Como manter o sincronismo dos relógios das máquinas de um sistema distribuído?
- Todas as máquinas possuem um temporizador que causa H interrupções por segundo. Quando o temporizador expira adiciona-se 1 no clock.
- Idealmente em uma máquina p:

$$C_p(t_{UTC}) = t_{UTC}$$

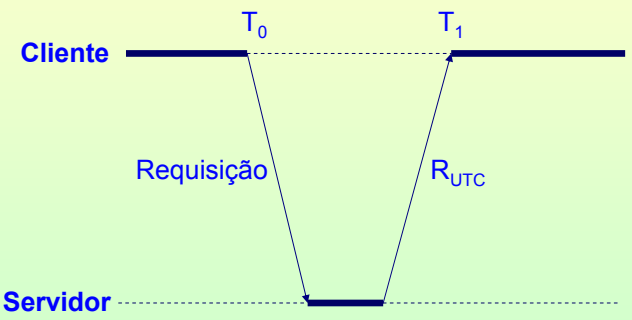
Algoritmo para sincronização de relógio



Algoritmo de Cristian

- Servidor de tempo: máquina sincronizada com um serviço que fornece o tempo UTC.
- Periodicamente cada máquina envia uma mensagem para o servidor de tempo. O servidor responde com o tempo corrente C_{UTC}.

Algoritmo de Cristian



Algoritmo de Cristian

- Problema: o tempo nunca pode andar para trás → mudança do tempo é feita gradativamente.
- Ao invés de se adicionar 10 ms a cada interrupção adiciona-se 9 ms até atingir o valor corrigido.
- Pode-se utilizar o mesmo esquema para adiantar o relógio.

Algoritmo de Cristian

- Outro problema: a requisição e a resposta gastam um tempo não nulo. E este tempo pode variar.
- Pode-se estimar o tempo que a mensagem gastou para retornar fazendo:

$$(T_1 - T_0) / 2$$

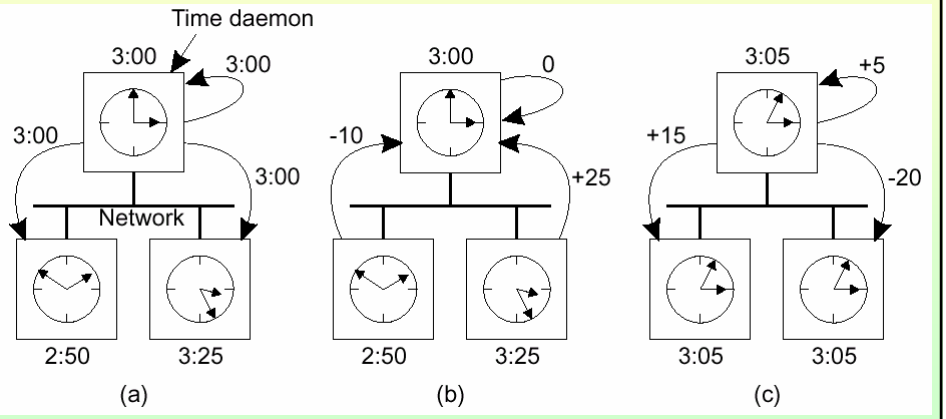
- Ou ainda, se soubermos o tempo que o servidor gasta para processar a requisição:

$$(T_1 - T_0 - I) / 2$$

Algoritmo de Berkeley

- Algoritmo adotado no UNIX Berkeley: se nenhuma máquina está sincronizada com o tempo UTC:
  1. o servidor de tempo consulta todas as máquinas
  2. calcula a média dos tempos
  3. informa a cada máquina se ela deve adiantar ou atrasar o relógio

Algoritmo de Berkeley



Algoritmo baseados na média

- Algoritmo descentralizado
- Divide-se o tempo em intervalos fixos de ressincronização (i-ésimo intervalo):
  - Início: $T_0 + iR$
  - Fim: $T_0 + (i+1)R$
    - → T_0: tempo conhecido por todas as máquinas
    - → R: parâmetro do sistema

Algoritmo baseados na média

- Em cada intervalo todas as máquinas devem:
  1. Enviar uma mensagem de broadcast com seu tempo corrente.
     - → as mensagens não vão ser emitidas ao mesmo tempo
  2. Iniciar um temporizador para coletar todas as outras mensagens.
  3. Calcular o tempo corrente com os valores coletados
     - → ex.: calcular a média dos valores podendo descartar os valores muito alto ou muito baixos

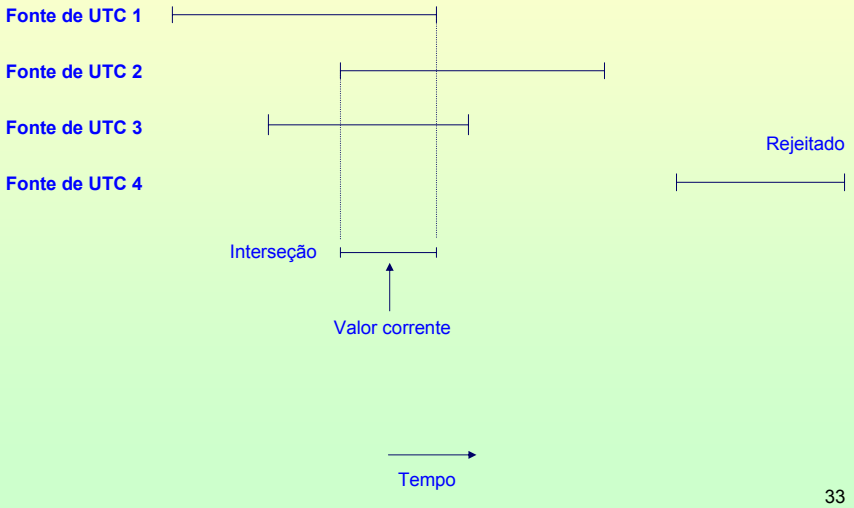
Sistema com várias fontes externas de tempo

- No caso de sistemas distribuídos onde seja necessária uma sincronização muito precisa pode-se utilizar várias fontes conectadas a serviços de UTC.
- Como calcular o tempo baseado nos vários servidores de tempo?
- Estabelece-se intervalos de valores válidos para o tempo UTC.

Sistema com várias fontes externas de tempo

- Sistema do DCE (Distributed Computing Environment) da OSF (Open Software Foundation):
  1. Cada servidor de tempo envia um broadcast periodicamente
  2. Cada máquina, quando obtém os intervalos de UTC, descarta os valores muito fora do esperado
  3. Calcula a interseção dos intervalos válidos
  4. Ajusta o seu relógio para coincidir com o ponto médio deste intervalo

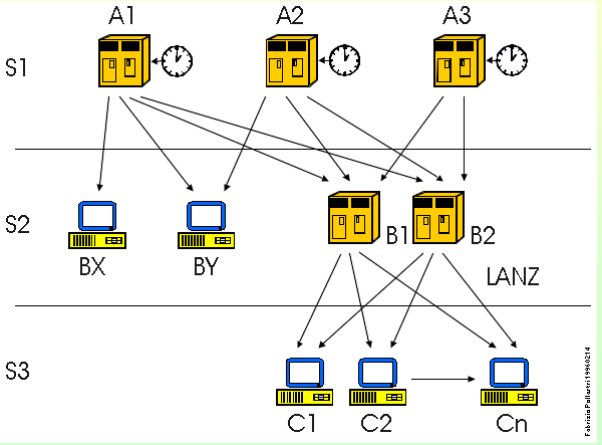
Sistema com várias fontes externas de tempo



NTP – Network Time Protocol

- Desenvolvido na Universidade de Delaware
- Características:
  - Usa o protocolo UDP e porta 123
  - Usa o algoritmo de Marzullo para determinar o valor do tempo
  - NTPv4 pode manter um relógio com precisão de 10 milissegundos na Internet e 200 microssegundos em uma LAN
  - Sistema hierárquico baseado em estratos (stratum)

NTP – Network Time Protocol



NTP – Network Time Protocol

- Página oficial:
  - http://www.ntp.org/
- Servidores do estrato 1
  - time.nist.gov
  - ntp1.rnp.br
  - ntps1.pads.ufrj.br

Exercícios

- “Sistemas operacionais modernos” Andrew S. TANENBAUM Prentice-Hall, 1995
  - Capítulo 11 Exercícios 1 - 3 (pág. 345)