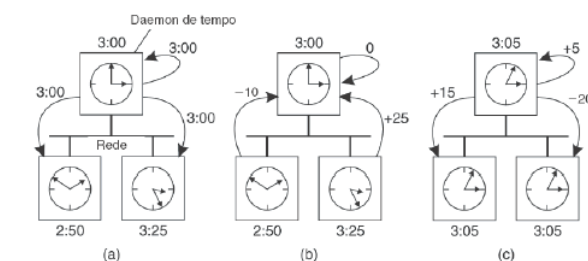


Considerando a ausência de relógio comum e global em um sistema distribuído, a sincronização de relógio pode ser realizada através de duas formas: física ou lógica. Um dos principais algoritmos de sincronização de **relógio físico** é o **Algoritmo de Berkeley** e, por sua vez, o **algoritmo de Lamport** é um dos mais importantes para sincronização baseada em **relógio lógico**. Com base no contexto acima, responda as seguintes questões:

1. Descreva brevemente a ideia e o funcionamento do algoritmo de Berkeley.
2. Sobre o algoritmo de Berkeley, cite uma vantagem e uma desvantagem (limitação).
3. Descreva brevemente a ideia e o funcionamento do algoritmo de Lamport.
4. Indique um tipo de sistema (cenário) que pode ser (ou é) beneficiada com a implementação e utilização de cada um destes algoritmos para sincronização de dados e recursos. Justifique sua resposta.

1) Algoritmo de Berkeley (relógio físico):

- Para sincronização interna em sistemas assíncronos.
- O Servidor de Tempo é ativo, e requer, periodicamente de cada máquina, o tempo do seu relógio.
- É escolhido um computador para ser o coordenador (master).
- O master periodicamente contata os outros computadores, (slaves).
- O master faz uma estimativa do tempo local de cada slave, baseado no RTT.
- O master calcula o tempo médio de todos os computadores (considerando o tempo dele mesmo), ignorando valores de transmissão demasiado elevados e máquinas com tempos muito diferentes.
- Finalmente o *master* envia a cada computador o valor de ajuste do seu relógio, podendo ser um valor positivo ou negativo.
- Caso o *master* falhe um novo computador será eleito para substituí-lo.



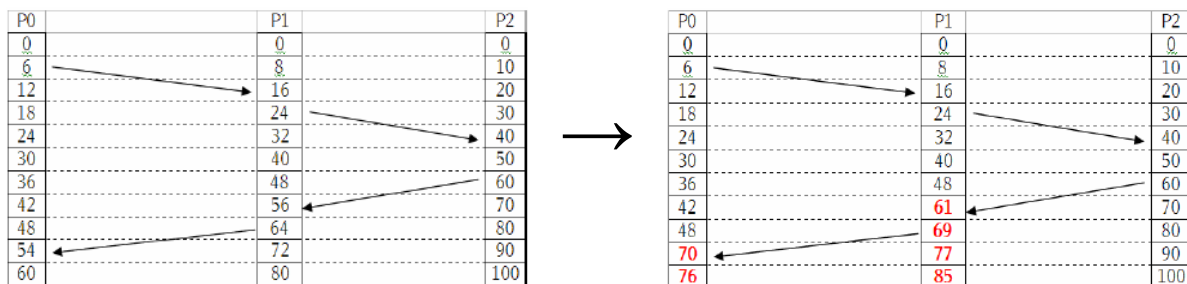
(a) O daemon de tempo pergunta a todas as outras máquinas os valores marcados por seus relógios.
(b) As máquinas respondem. (c) O daemon de tempo informa a todas como devem ajustar seus relógios.

2) Algoritmo de Berkeley - vantagem e desvantagem:

Não existe um tempo global em SDs, pois cada computador tem seu próprio relógio físico, estes relógios não são exatos, desviam ao longo do tempo, sendo uma desvantagem não se conseguir sincronizá-los perfeitamente. A vantagem é existir o tempo real de execução, baseada na média após cada sincronização, aferindo qualidade a este tempo.

3) Algoritmo de Lamport (relógio lógico):

- Sincronização baseada em “tempo relativo”.
- Tempo relativo não possui nenhuma relação com o tempo real.
- Se dois processos não interagem, não é necessário que seus *clocks* sejam sincronizados.
- Usualmente o que importa não é que todos os processos concordem com o exato tempo em que os eventos aconteceram, mas que concordem na ordem em que os eventos ocorreram.
- Lamport definiu a seguinte relação: “acontece-antes”:
 - $a \rightarrow b$ (**a** acontece antes de **b**)
- Significa que todos os processos concordam que primeiro o evento **a** ocorreu e depois disto, o evento **b** ocorreu. Esta relação pode ser observada em duas situações:
 - Se **a** e **b** são eventos no mesmo processo, e **a** ocorre antes de **b**, então $a \rightarrow b$ é verdadeiro.
 - Se **a** é o evento de uma mensagem sendo enviada por um processo, e **b** é o evento da mensagem sendo recebida por outro processo, então $a \rightarrow b$ é também verdadeiro.
- A relação “acontece-antes” é transitiva, logo:
 - Se $a \rightarrow b$ e $b \rightarrow c$, então $a \rightarrow c$
- Se dois eventos **x** e **y**, acontecem em diferentes processos que não trocam mensagens, então $x \rightarrow y$ não é verdadeiro, nem $y \rightarrow x$ é verdadeiro.
- Estes eventos são ditos concorrentes, o que significa que nada pode ser dito sobre quando eles aconteceram.
- É necessário um modo de medição de tal forma que para todo evento **a**, possa ser assumido que ele ocorreu em um tempo **C(a)** no qual todos os processos concordem.
 - Se $a \rightarrow b$, então $C(a) < C(b)$.



4) Cenário hipotético para o algoritmo de Berkeley e de Lamport:

A escolha de cenário depende dos processos que compõem a aplicação, ou seja; é mais importante saber a ordem de ocorrência de um processo ou o tempo real de sua execução? No caso de um sistema bancário pode-se utilizar o algoritmo de Lamport, haja vista a ordem dos processos ser mais significativa do que seu tempo exato de ocorrência. Já noutra aplicação em que a sincronização é fundamental, visando o mais próximo do tempo real, como num sistema de controle de tráfego aéreo, o algoritmo de Berkeley será a escolha mais eficiente.