

# Projeto: Algoritmo de Berkeley

O algoritmo de Berkeley permite sincronizar diversos computadores em um sistema distribuído. Para o entendimento do problema e da solução apresentados originalmente, veja o trabalho publicado "*The accuracy of the clock synchronization achieved by TEMPO in Berkeley UNIX 4.3BSD*", em <https://doi.org/10.1109/32.29484> . E, se precisar de mais alguns exemplos, veja: [https://lasarojc.github.io/ds\\_notes/time/physical/#algoritmo-de-berkeley](https://lasarojc.github.io/ds_notes/time/physical/#algoritmo-de-berkeley)

Além disso, esse algoritmo também pode ser visto nos capítulos dos livros adotados na disciplina: para isso, veja o Plano de Ensino disponibilizado.

Neste projeto, considere inicialmente um sistema distribuído responsável pelo controle de compra/venda de ações em uma bolsa de valores (BV). Essa BV detém uma lista de ações, a quantidade dessas ações disponíveis para compra/venda e o valor atual de cada ação. Nessa simulação, para cada operação de compra, o valor da ação sobe pelo próprio sistema da BV; para cada venda, o valor da ação desce pela BV.

Todas as vezes que a quantidade ou o preço de uma ação é modificado, essas informações são propagadas para as instituições intermediárias e seus sistemas de *home-brokers* (HB). Os sistemas de HB são responsáveis pelo envio dos pedidos de compra/venda para a BV. Na mensagem de compra/venda também está registrado a hora do sistema de HB.

Ao analisar o pedido de compra/venda do HB e a hora em que ele foi emitido, a BV pode decidir por acatar o pedido ou pode decidir por iniciar um processo de sincronização usando o algoritmo de Berkeley – todos os sistemas de HB terão suas horas ajustadas conforme esse algoritmo. Lembre-se que para acatar o pedido os valores das ações devem ter algum grau de coincidência.

O pedido de compra/venda das ações é feito por um sistema de robô que decide se uma ação deve ser comprada ou se uma ação que ele possui deve ser vendida. Esse pedido é enviado pelo robô para um HB no qual o robô está vinculado. O robô só registra a compra/venda após a confirmação de decisão feita pela BV.

Para essa simulação, considere um sistema distribuído que tenha:

- mínimo de 1 processo representando a BV
- mínimo de 2 processos de HB vinculados a BV;
- mínimo de 2 robôs para cada HB existente;
- o relógio de cada processo, no início do seu funcionamento, deve ser recuperado do relógio do sistema;
- a cada ciclo de 10 segundos, o relógio local de cada processo deve ser modificado aleatoriamente em  $\pm 2s$  – faça o controle lógico do relógio real baseado nessas modificações, uma vez que não teremos permissão para alterar os relógios físicos dos computadores de laboratório;
- ao receber uma mensagem, a bolsa de valores analisa o tempo de quando a mensagem foi enviada e se o tempo foi maior que 1s do seu relógio local, ela inicia o processo de sincronização.

Pontos de atenção:

- não precisa considerar falha de mensagens;
- o processo de sincronização de relógio sempre acontecerá independentemente da concorrência de troca de mensagens enviadas de mais de um processo;
- cuidado com a compra/venda das ações para que não haja perda de informação: ao final do expediente (ou simulação), a quantidade de ações deve ser a mesma do início do expediente;
- faça o registro/log das ações dos processos: início da sincronização: hora dos processos antes da sincronização; hora depois da sincronização; etc;
- seu código deve ser desenvolvido em Java, Python ou C/C++ - preferencialmente, misturando as 3 linguagens;
- é obrigatório o uso de computadores diferentes para visualizar o teste em ambientes reais – pode-se utilizar também máquinas virtuais no seu provedor de nuvem predileto ou até mesmo local.