

- 1) O problema de sincronização de relógios é que se é requerido pelo sistemas constantemente e as vezes de forma estritamente precisa, dessa forma, podendo ocasionar de ser gerado um código em um computador e ser executado em outro computador que esteja com seu horário atrasado e faça com que os arquivo seja executado antes mesmo dele ser construído se for analisar os horários da forma em que estão dispostos, e assim ocasionando inconsistência no sistema.
- 2) 1- Não existe relógio global;  
2- Atrasos nas mensagens trocadas;  
3- Relógios podem dessincronizar com o tempo.
- 3) Relógio físico - Ou melhor chamado também de temporizador, é um circuito composto na maioria dos computadores onde é monitorado a passagem do tempo, esses temporizadores trabalham sobre uma frequência e tensão específica podendo mudar de máquina para máquina.  
Relógio Lógico - O que realmente importa não é que todos os computadores estejam concordando com o mesmo horário de forma exata e sim a ordem em que os eventos ocorrem.  
A principal diferença entre os dois é que um contabiliza a hora propriamente dita e o relógio lógico contabiliza a cada evento que acontece, assim não dependendo de uma hora igual em todos os computadores.
- 4) O algoritmo de Berkeley usa de servidores de tempo, esse servidores recebem a hora de algumas máquinas que são selecionadas para que se garanta uma captação de horário não muito distante do que é preciso, depois de pegar todas as horas das máquinas é feito uma média dessas horas e é dado uma resposta para cada computador sincronizar com essa média seja adiantar ou atrasar os relógios das suas respectivas máquina.
- 5) 1- O sistema se comunica a outras máquinas de outra rede.  
2- O atraso das mensagens entre o servidor e os computadores.
- 6) O algoritmo de Lamport para cada processo denominado de  $P_i$  tem um contador denominado de  $C_i$ , onde esse contador é incrementado a cada evento disparado em  $P_i$ , assim se dois processos interagem entre si, não tem necessidade de seus relógios estarem sincronizados. Esse relógio tem pode ser conhecido como ordenação casual potencial, pois um processo a pode ter tempo menor do que um processo b, pois casualmente quem mandou o sinal para b começar foi a, assim b depende casualmente de a.
- 7) Funciona como um vetor de relógios de Lamport, ou seja, cada processo é identificado por um vetor, assim a única diferença entre os relógios vetoriais e os relógios de Lamport é que os vetoriais a cada evento disparado em um processo é incrementado apenas o relógio lógico referente a posição do processo guardando assim separadamente o tempo de cada processo, dessa forma, diferente dos relógios de Lamport, não tem como haver conflito de tempo por serem medidos todos em um local só, ou seja, o vetorial guarda o valor de cada processo separadamente diferentemente do de Lamport que possui todos os valores em um único local.
- 8) Dependendo do sistema e da quantidade de recurso que está sendo utilizada, temos que dois processos precisam usar o mesmo espaço de memória ocasionando assim uma disputa entre processos para conseguirem usar o espaço de memória.
- 9) Trata-se de um método de assegurar que se um processo já está usando um determinado recurso, outros processos não poderão acessar o mesmo recurso

enquanto o recurso não estiver disponível. A importância dessa técnica é que não ocasionará mais a disputa por um recurso, ou seja, dois processos não tentaram usar o mesmo recurso ao mesmo tempo.

- 10) Algoritmo centralizado - Existe um coordenador único, ou seja, as desvantagens desse algoritmo é que existe apenas um ponto de falha, esse ponto pode ser um ponto de gargalo do sistema e também fica inviável vê se coordenador está inativo ou apenas a permissão ao recurso está negada.

Algoritmo descentralizado - Existem vários coordenadores, cada recurso é replicado n vezes e cada réplica de recurso tem o seu próprio coordenador, assim quando um processo deseja acessar um recurso ele precisa de um voto majoritário, e o coordenador informa ao processo se ele tem permissão para acessar aquele recurso ou não, os principais problemas é que se muitos nós requisitarem acessar o mesmo recurso, a utilização do recurso decresce rapidamente e também se houver muitos processos competindo acesso, haverá um momento em que nenhum deles conseguirá votos o suficiente e o recurso deixará de ser utilizado.

- 11) Algoritmo Token Ring - Processo são organizados em uma topologia anel, onde o processo recebe uma ficha ou token para poder utilizar o recurso, assim a ficha ou token é enviado de processo em processo e ela só ser enviada pela sequência do anel e os processos só conhece seu vizinho da sequência, assim o processo deseja usar o recurso ele fica com a ficha até concluir a execução, caso não deseja utilizar o recurso ele passa a ficha para o próximo processo, as desvantagens desse algoritmo são atrasos na utilização do recurso, o fato da ficha sempre passar mesmo que nenhum processo esteja precisando daquele recurso e dificuldade em saber se o vizinho a receber a ficha morreu ou não, podendo desconfigurar o anel.

Algoritmo distribuído - Requer que ordenação total dos eventos no sistema para saber qual processo solicitou primeiro o recurso, assim, quando um processo deseja acessar um recurso compartilhado, monta uma mensagem que vai ser enviado para todos os processos contidos no sistema contendo o nome do recurso, seu número de processo e a hora corrente, após enviar a mensagem a todos os processos, se o receptor da mensagem não tiver acessando o recurso ou não quiser nem acessar ele retorna ok para o processo que enviou a mensagem, caso o processo já possui acesso ao recurso ele não responde nada e colocar o processo que enviou a mensagem na fila, dessa forma, ao enviar as mensagens ele espera a resposta de todos e assim quando houver a permissão de todos ele acessa o recurso e assim que tiver terminado o acesso ao recurso ele envia um ok a todos os outros processos.

- 12) Algoritmo da eleição - é responsável por escolher qual o processo desempenha a função de servidor em uma exclusão mútua centralizada, assim o algoritmo seleciona qual será o coordenador único do algoritmo centralizado.

O algoritmo considera todos os processos exatamente iguais como também cada processo contém um id de identificação e um pid que diz o desempenho do processo, como também cada processo conhece o número exclusivo dos demais mas não sabe se estão ativos ou não.

- 13) O requerente do recurso envia uma mensagem de requisição e inicia-se um temporizador, caso atinja o tempo máximo sem a resposta a concessão do recurso inicia-se uma eleição pois quem comandava as requisições é presumido que ele foi desativado. Esse algoritmo tem três mensagens: a de eleição que é utilizada para

convocar a eleição, a de resposta utilizada para encerrar uma eleição e a de coordenador utilizada para anunciar o processo eleito.

- 14) Caso exista um processo com maior identificador do que o que iniciou a eleição, quem iniciou a eleição termina a sua eleição e quem mandou mensagem dizendo que possuía maior identificador inicia sua própria eleição.

Caso todos os processos tenham identificação menor ao processo que iniciou a eleição, o processo que fez a eleição termina a eleição e se torna o coordenador mandando mensagem a todos os outros processos informando-os.