

Nome : Lays de Freitas Melo 201801514

Nome : Allyson Alvim Ribeiro 201702724

Nome : Felipe Gabriel Fraga Pinto 201702739

Relatório Exclusão - Mútua

1. Algoritmos do tipo exclusão mútua, que compartilham a mesma meta fundamental: um conjunto de processos coordenar suas ações ou concordar com um ou mais valores. Exemplo da nave.
2. Computadores que estão no comando devem concordar com as condições de estado.
3. Coordenar corretamente ações com relação aos recursos compartilhados.
4. Ser capazes de fazer de fazer essas tarefas mesmo onde não haja nenhum relacionamento mestre-escravo fixo entre os componentes.
5. O motivo de se evitar relacionamentos mestre-escravo fixos é que, frequentemente, exigimos que nossos sistemas continuem funcionando corretamente, mesmo que ocorram falhas; portanto precisamos evitar pontos de gargalo, no caso os mestres fixos.
6. Uma distinção importante é se o sistema é assíncrono o síncrono. Em um sistema assíncrono não é possível realizar suposições quanto à temporização. Em um sistema síncrono devemos supor que existem limites para o atraso máximo na transmissão das mensagens, para o tempo que leva para executar cada etapa de um processo e para as taxas de atraso.
7. Suposições em um sistema síncrono nos permite usar tempos limites para detectar falhas de processo.
8. Considerar falhas e como lidar com elas ao projetar algoritmos.
9. Mesmo sob condições de falhas benignas, em um sistema assíncrono é impossível garantir que um conjunto de processos possa concordar com um valor compartilhado
10. Problema da exclusão mútua distribuída. Trata-se da ampliação para os sistemas distribuídos do familiar problema de evitar condições de disputa nos núcleos e em aplicativos multithreadeds.
11. Como “eleger” um processo a partir de um conjunto de processos, para desempenhar uma função especial (Eleição).
12. Se um servidor (“referência”) falhar e vários servidores sobreviventes puderem desempenhar essa função, então, por causa da consistência, é necessário escolher apenas um servidor para assumir o comando.
13. capacidade de fazer multicast de uma mensagem para um grupo é um paradigma de comunicação muito útil, com aplicações desde a localização de recursos até a coordenação das atualizações em dados replicados.
14. A entrega de mensagens baseada em multicast é basicamente um problema de acordo entre processos: os destinatários concordam com quais mensagens receberão e em que ordem as receberão.
15. Por simplicidade, presume-se que cada par de processos está conectado por canais confiáveis. Isto é, embora os componentes de rede subjacentes possam falhar, os processos usam um protocolo de comunicação confiável que mascara essas falhas.
16. Os processos precisam ser independentes, para que nenhuma falha seja uma ameaça para o funcionamento de outro processo.
17. Em cada processo existe um objeto local chamado de detector de falha local. A maioria dos detectores de falha caem na categoria de não confiáveis. Um detector de falha não confiável pode produzir um de dois valores: suspeito ou não suspeito que podem refletir sobre o real estado do processo.
18. Um detector de falha confiável é aquele que é sempre preciso na detecção da falha de um processo.
19. Em um sistema síncrono, um detector de falha pode se tornar confiável.