Lista de Exercícios não-exaustiva p/ P1 de SOD

1. Descreva dois tipos de aplicações onde sistemas operacionais distribuídos são preferidos em relação aos sistemas centralizados? Explique por que.

2. Um servidor funciona cerca de 85 % do tempo e o restante fica fora do ar para manutenção ou por problemas técnicos. Quantas vezes este servidor deve ser replicado de forma que o serviço que ele atende 99,99 % no ar?

3. Descreva a principal diferença entre comunicação com sockets e RPC.

4. Compare as principais características de um sistema operacional microkernel e de um sistema operacional monolítico.

5. Em que contexto a troca de mensagens aplica-se melhor?

6. Faça uma comparação entre o funcionamento de um send bloqueante e de um send não-bloqueante.

7. Explique o send implícito. Para que serve?

8. Com relação à implementação dos buffers de espera:

    - Qual deles se adaptaria idealmente a um sistema síncrono de troca de mensagens?
    - Como funciona o Finite Bounded Buffer?
    - É possível implementar o Unbounded capacity Buffer? Explique sua resposta.

9. Identifique os três tipos possíveis de falhas que podem comprometer o funcionamento de um esquema de comunicação baseado em troca de mensagens.

10. Com relação ao tratamento de falhas na troca de mensagens:

    - Qual o esquema mais confiável?
    - Qual esquema apresenta maior overhead de comunicação?

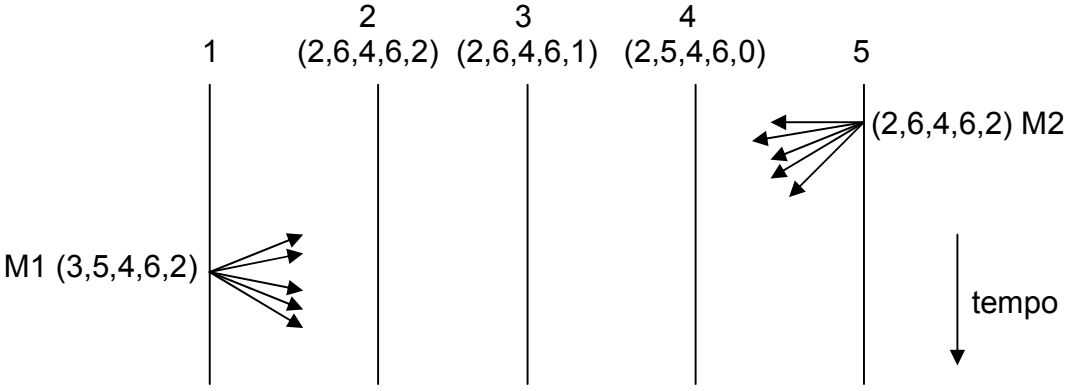
11. Que tipo de falha o esquema baseado em três mensagens não consegue identificar?

12. No processo de comunicação de grupo a ordem de entrega das mensagens é muito importante, existem diversas formas de ordenação para entrega de mensagens. O algoritmo CBCAST implementa ordenação de causa. Descreva se este algoritmo pode ou não ser utilizado para implementar ordenação consistente. Justifique sua resposta.

13. Considere um sistema que possui um servidor aritmético que executa as operações soma, subtração, multiplicação e divisão. O sistema utiliza um mecanismo RPC e para cada requisição recebida o servidor cria um processo para executá-la e volta a esperar por uma nova requisição. Com o uso de sockets para comunicação, escreva o programa servidor. Escreva também um programa cliente.

    Primitivas:
    sock = socket (familia, tipo, protocolo)
    newsock = accept (sock , ...)
    connect (sock , ...)
    listen (socket, quantos)
    bind (socket, ...)
    send (socket, &data, sizeof(data))
    receive (socket, &data, sizeof(data))

14. Imagine um grupo com 5 processos (1, 2, 3, 4, 5) com os vetores contendo os valores mostrados na figura abaixo. O processo 1 envia uma mensagem M1 para todos os demais processos do grupo com o seu vetor já atualizado, e o processo 5 também envia mensagem M2 para os demais processos, com seu vetor já atualizado. Quais dos processos podem receber as mensagens, quais devem armazenar as mensagens até que outras cheguem e quais já receberam as mensagens? Mostre por que. Verifique se existe inconsistência nos sistema.



15. Descreva os passos executados quando um processo chama um procedimento remoto usando RPC.