

Nome: _____ Matr.: _____

1) a) Quais as vantagens da sincronização de processos usando troca de mensagens sobre as sincronizações usando as técnicas vistas no início do curso como as técnicas de monitores ou semáforos?

b) Na internet existem diversos roteadores que podem ser ligados e desligados, sem que isso interfira, de forma significativa, no funcionamento da rede. Um roteador que acaba de ser ligado deve anunciar a sua presença, enviando uma mensagem especial (HELLO), contendo a sua identificação (endereço IP e nome lógico) para cada uma de suas interfaces e aguarda uma mensagem de resposta. Para cada interface na qual ele recebe uma mensagem de resposta, o roteador assume que tem um vizinho ativo. Para as interfaces nas quais ele não recebe mensagem de resposta ele considera que não existe vizinho ativo. Que tipo de comunicação (síncrona ou assíncrona) e que esquema de nomeação (implícita ou explícita) devem ser utilizados ? Justifique.

2) Considere os processos (tasks) **proc1** e **proc2** mostrados a seguir.

task body proc1 is a : integer ... “calcula o valor de a” proc3 .param_1(a) ...	task body proc2 is b : integer ... “calcula valor de b” proc3 .param_2(b) ...
--	--

task body proc1 is a : integer ... “calcula o valor de a” proc3 .param_1(a) ...	task body proc2 is b : integer ... “calcula valor de b” proc3 .param_2(b) ...
--	--

a) Esboce o código da *task* **proc3** – cabeçalho e corpo – detalhando os comandos de comunicação no estilo ADA.

A solução apresentada deve atender para a task **proc3** deve atender aos seguintes requisitos:

- permitir a recepção de um único valor de cada task (**proc1** e **proc2**);
- a ordem de recepção deve ser qualquer;
- com os 2 valores recebidos a task **proc3** deve verificar qual o maior valor recebido e repassá-lo para uma outra task de nome **proc4** usando um ponto de entrada (procedimento do tipo *entry*) de nome **maior**.
- **NÃO** é necessário mostrar o código de **proc4**.

b) Mostre as alterações que devem ser feitas no corpo da task **proc3** no caso das tasks **proc1** e **proc2** serem transformadas em loops infinitos.

A cada interação na task **proc3** deve ser recebido um único valor de **proc1** e um único valor de **proc2** e apenas o maior deverá ser repassado para **proc4** (que também será um loop infinito).

- 3) Considere o problema do jantar dos canibais. Os processos canibais requisitam um pedaço de carne a um processo Pote. O processo Pote, a cada mensagem recebida de um processo canibal, decrementa o número de pedaços de carne e responde com outra mensagem para o processo canibal (informando que ele pode comer). Quando um canibal requisita o último pedaço de carne, o processo Pote envia uma mensagem a um processo Cozinheiro, que retorna uma resposta informando um novo número de pedaços de carne disponível (a cada vez que o Cozinheiro é chamado, ele pode informar um número diferente). Suponha que inicialmente o Pote possui 10 pedaços de carne. Implementa uma solução com 10 processos canibais usando troca de mensagens assíncronas.
- 4) MPI

Boa Prova! Boas Férias e Boas Festas!!!