

2ª Série de
CE-288/CES-27 Programação Distribuída
Setembro de 2011

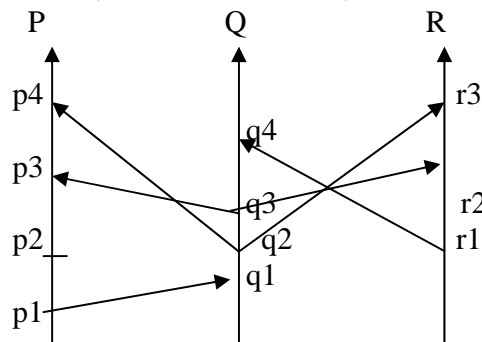
Primeira Parte: Java

1. Dê exemplo de como implementar os métodos *suspend()* e *resume()* para Threads sem usar os métodos que foram “deprecated” (entraram em desuso).
2. Por quê o código abaixo deve estar dentro de um método sincronizado?

```
while (!condition)
    wait();
```
3. O que são applets? Quais as restrições de uma applet em execução? Qual é a hierarquia da classe *Applet*? Quais os métodos de *Applet* vistos em aula?
4. Como se faz para tratar eventos de mouse dentro de objeto *Applet*?
5. O que se deve ter em código Java no servidor e no cliente para estabelecer uma comunicação em TCP/IP? Dê um exemplo.
6. O que se deve ter em código Java no servidor e no cliente para estabelecer uma comunicação em UDP? Dê um exemplo.
7. Descreva as declarações que são necessárias para se ter uma implementação RMI. Justifique a descrição.

Segunda Parte: Algoritmos

1. Descreva uma implementação do comando **SELECT RECEIVE** do exemplo **BOUNDEDBUFFER** na apostila em Java.
2. Complete o algoritmo de Misra (ping-pong) visto em aula. Explique com um exemplo porque não é permitida a ultrapassagem neste algoritmo.
3. Complete a resolução do exercício do algoritmo de Exclusão Mútua de Lamport. Explique com um exemplo porque não é permitida a ultrapassagem no algoritmo do Lamport.
4. Rotule o diagrama abaixo com:
(a) valores dos relógios lógicos (inicialmente 0)
(b) Tempo de Vetor (inicialmente [0,0,0]).



5. Suponha a seguinte transação de banco de dados distribuídos: transferência de R\$ 100,00 de uma conta A do Banco Alfa para uma conta B do Banco Beta. Suponha que os bancos têm seus sistemas instalados em cidades diferentes e a transação está sendo feita de uma terceira cidade e a conta A está com um saldo de R\$ 300. (a) Descreva as operações de leitura e gravação necessárias nos dois locais. (b) Descreva a troca de mensagens usando Protocolo de Cometimento de Duas Fases para uma transação com sucesso. (c) Mostre o conteúdo dos arquivos *log* para uma transação com sucesso. (d) Mostre o conteúdo dos arquivos *log* para uma transação com uma falha do *host* do Banco Beta depois de gravar o *ready*.
6. Verifique se as execuções distribuídas abaixo são serializáveis e caso elas sejam dê a seqüência das transações.
- a) Execução 1: S1: $R(1,a) < R(3,d) < W(2,a)$
 S2: $W(2,b) < R(1,e) < R(3,b)$
 S3: $W(3,c) < R(2,f) < R(2,g) < R(1,c)$
- b) Execução 2: S1: $R(2,b) < R(2,c) < R(1,a) < W(1,b) < R(1,c) < R(3,c) < W(3,a)$
 S2: $R(2,d) < R(2,e) < R(3,e) < W(1,e) < W(3,d)$
 S3: $R(2,g) < R(2,h) < W(2,h) < W(1,g) < W(3,g)$

Observações:

- A entrega da série é **obrigatória**, e ela deve ser **entregue** antes da prova, uma vez que o objetivo é **preparar** o aluno para a prova.
- As questões da prova deverão ser **baseadas** na série.
- A prova é **Sem Consulta**.
- A prova será no dia 3 de outubro (Disciplina Isolada e Mestrado) e 10 de outubro (Graduação), às 9h.