INE5645 – Programação Paralela e Distribuída – Turma B - Prof. Bosco - 07/07/2014

Nome \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Indique Verdade (V) ou Falso.

Cada questão rigorosamente correta vale 1 ponto.

1. ( **F** ) Sockets TCP *stream* utilizam o protocolo (TCP) **UDP** com **datagramas** (fluxo de bytes).
2. ( **V**  ) RMI utiliza sockets entre os módulos de comunicação de baixo nível, seguindo o modelo cliente-servidor com objetos distribuidos. RMI é um modelo de programação e sockets implementam comunicação assíncrono ou síncrono (TCP ou UDP, respectivamente.
3. ( **V** ) Importância de transparência em RMI - Todas as etapas necessárias para empacotamento e troca de mensagens, mais a tarefa de localizar e contactar um objeto remoto, são ocultadas do programador que faz uma chamada remota. Em um sistema distribuído com transparência de localização, o cliente não precisa saber da localização do servidor.
4. ( **F**  ) Em RMI, o objeto que está fazendo a invocação **não** pode identificar se o objeto é local ou remoto.
5. ( **V** ) Middleware é uma camada de software que fornece um modelo de programação (apropriado para aplicativos distribuídos, ou seja, para implementação de aplicações com objetos distribuídos, compostos de programas que estão em cooperação, executados em várias máquinas), e que é situado entre camadas de software (no nosso caso, o Java RMI, entre as camadas de Aplicação e de Transporte usa um protocolo baseado em mensagens entre objetos (locais e remotos), no sentido de fornecer abstrações de mais alto nível (o que é de baixo nível fica oculto pela transparência de localização), como as invocações remotas, proporcionando transparência de localização nas invocações remotas.
6. ( **V**  ) **Protocolos de comunicação em middleware**: Os protocolos que suportam as abstrações/funcionalidades de um *middleware* são independentes dos protocolos de transporte subjacentes. Esses protocolos geram mensagens em código binário. Assim, são protocolos que devem utilizar mecanismos de segurança contra ataques situados nas mensagens em binário. Isto implica que RMI na Web, não é apropriado.

Como hoje, o mundo todo se comunica por HTTP, que passa por todos os firewalls, RMI ficou com uso mais restrito.

1. ( **V** ) Interfaces remotas e referências de objetos remotos caracterizam o modelo de objetos distribuídos.
2. ( **V**  ) A noção de **referência de objeto** é estendida para permitir qualquer objeto que possa receber uma invocação remota, tenha uma **referência de objeto remoto**. Essa referência é um identificador que pode ser usado por todo o sistema de objetos distribuídos, para se referir a um objeto único em particular. Sua representação contém um **endereço IP**, **número de porta**, **hora**, **número do objeto local** e **interface do objeto remoto**. Juntos, o **número de porta** e **hora,** produzem um identificador do objeto. O objeto remoto que vai receber uma invocação num método remoto, é especificado pelo invocador como uma referência de objeto remoto . As referências de objeto remoto podem ser passadas como argumentos e resultados de invocações a métodos remotos. Em Java RMI, um objeto-cliente precisa, para invocar um método remoto, conhecer uma referência de objeto remoto, a qual é usada no instante da invocação.
3. Que vantagens você pode citar do uso de RMI para desenvolver aplicações de objetos distribuídos em rede, em relação ao uso de sockets ?
4. Transparência de localização é provida pela camada de RMI, quando da procura pela localização de objetos de uma aplicação distribuída numa rede. A camada RMI funciona como um ORB (Object Request Broker), que é como um corretor de imóveis que procura e localiza imóveis. Um ORB localiza objetos. Veja a questão 3.
5. RMI implementa em mais alto nível, enquanto socket é uma tecnologia de mais baixo nível. RMI, por sua vez, usa sockets para a sua comunicação. Sockets cria um canal de comunicação onde você próprio tem que desenvolver o protocolo. Sockets estão por baixo de tudo. Por exemplo, O Tomcat e JBoss usam sockets, É fundamental saber usar e também aprender sobre o protocolo TCP/IP, pois sockets é um conceito de redes TCP/IP. Em RMI, esse protocolo já está implementado e você não vê os sockets.
6. RMI não significa que seja somente Java. Existem outros RMI, por exemplo, o modelo de programação do CORBA, que aceita utilizar outras linguagens, por exemplo, um cliente Java e um servidor C++. Mas, volte lá na questão 6.
7. RMI em Java, incorpora segurança do modelo de segurança do Java. Sockets exige que, se desejada algum nível de segurança, esta seja implementada.

1. Uma interface de *Eleição* provê duas operações:

***vote*:** com dois parâmetros através dos quais o eleitor fornece o nome de um

candidato (uma string) e o número do eleitor (um inteiro uado para garantir que cada

cliente vota somente 1 vez). Os números dos eleitores são alocados, espaçadamente,

dentro de um domínio de inteiros para torná‐los difícil de adivinhar.

***result*:** com dois parâmetros através dos quais o servidor fornece ao apurador o nome

de um candidato e o número de votos obtidos para aquele candidato.



1. Qual (verifique na tabela) a semântica de invocação que pode ser utilizada no **Java RMI**, para o funcionamento correto (um votante não pode votar duas vezes) do sistema de votação eletrônica especificado ?

Semântica de invocação = No máximo uma vez

1. Diga, verificando na tabela, que medida de tolerância à falha seria utilizado na semântica escolhida por você, caso houvesse perda da conexão entre a execução do voto e a reposta ***result*** solicitada.

Retransmissão da Resposta