

Pergunta: Defina CORBA e de um exemplo de uso

Resposta: CORBA, ou Common Object Request Broker Architecture é um padrão definido pela OMG (object management group), que define uma arquitetura para componentes de software de diferentes linguagens e sistemas se comuniquem. Ele usa uma IDL (interface definition language) para usar um ORB (object request Broker) que realizava a comunicação sobre um protocolo IIOP. Um exemplo de utilização do CORBA é o sistema usado pelo weather Channel para fazer a comunicação de suas centrais.

Pergunta: Qual o principal objetivo do IIOP e como ele atinge esse objetivo?

Resposta: O principal objetivo do IIOP é a interoperabilidade entre ORBs concorrentes. Essa interoperabilidade é alcançada através do mapeamento para TCP/IP do General Inter-ORB Protocol ou GIOP. O IIOP permite que requisições sejam enviadas para objetos distribuídos gerenciados por outros ORBs em outros domínios.

Pergunta: Qual a função de uma RFP para o CORBA?

Resposta: RFP é o modo pelo qual o OMG complementa o CORBA, adicionando a ele funcionalidades que são consideradas necessárias pela indústria. É um trabalho conjunto dos membros do OMG, que propõem a RFP, implementam soluções, criam diretivas, etc

Pergunta: Qual o objetivo do POS (Persistent Object Service)?

Resposta: Prover interfaces comuns para os mecanismos usados no armazenamento e gerência de estados persistentes de objetos.

Pergunta: Qual a principal função do ORB?

Resposta: Dele é a função de intermediar todas as transferências entre cliente e servidor, e fazer com que a transação seja transparente para cada uma das partes durante todo o processo. É responsável pela localização do objeto ao qual se destina a requisição, assim como o envio dos parâmetros da requisição no formato aceito por este objeto.

Pergunta: Como se dá o acesso a um OODB (Object-Oriented Database) em CORBA?

Resposta: Através de um serviço POS (Persistent Object Services) chamado ODBMS (Object Database Management System), que provê armazenamento persistente de objetos. O ODBMS

controla o acesso concorrente aos objetos, protege-os contra ameaças e toma conta de tarefas como backup e restore.

Pergunta: O que faz o POM e qual sua importância na arquitetura do POS?

Resposta: O POM (Persist Object Manager) provê uma interface uniforme para a implementação das operações de persistência de um objeto, independente do PDS utilizado.

Ele é extremamente importante, pois é devido à existência do POM e de um protocolo de comunicação entre PO e PDS que é possível trocar arbitrariamente o PDS que acessará o datastore, de forma "plug-and-play".

Pergunta: O que é DCOM? Como ele faz sua comunicação?

Resposta: DCOM, ou Distributed Component Model, é uma tecnologia privada da Microsoft desenvolvida a partir do COM (Component Object Model), e que realiza a integração diferentes objetos de maneira remota. Atualmente substituída pela implementação .NET remoting, que é mais simples.

Ao contrário do CORBA, o DCOM só funciona em arquiteturas windows. Ele se comunica através de um código único para cada função chamado CLSID, e dois objetos, um proxy e outro stub, no cliente e servidor respectivamente. Os identificadores encontram os outros objetos através de SCMs e registros guardados fora do cliente. As mensagens devem ser pasadas para usando o run-time system do COM. Elas são passadas através de um canal RPC.

Pergunta: O que é DSI e DII e como essas interfaces se relacionam para disponibilizar o método para um dado objeto CORBA ?

Resposta: DSI ou Dynamic Skeleton Interface, é um mecanismo que nos permite enviar requisições para uma ORB sem conhecer sua implementação de antemão. Podemos acessar um método sem ter um proxy determinado. Isso se faz através do DII, ou Dynamic Invocation Interface, que permite a invocação dinâmica de objetos. Assim, o DII pega as informações necessárias do objeto desejado e o DSI permite aceitar invocações sem esqueletos, devido ao fato de não termos nenhuma informação da interface em tempos de compilação.

Pergunta: Qual é a diferença entre GIOP e IIOP?

Resposta: GIOP é um protocolo abstrato que os object request broker(ORBs) se utilizam para se comunicar, tendo cada um sua implementação específica. Já IIOP é uma implementação de GIOP para TCP/IP que é utilizada num desses

ORBs, mais especificamente o CORBA.
IR

Pergunta: Para que é usado o IR?

Resposta: É um componente em tempo de execução na arquitetura ORB que é usado para dinamicamente obter informação dos tipos IDL. Esse tipo de informação é necessário quando um cliente quer usar a Dynamic Invocation Interface (DII) para construir pedidos dinamicamente, sem conhecimento estático sobre o tipo do outro objeto.

Nesse caso, quando não há informações disponíveis sobre um objeto CORBA, o IR pode ser requisitado a retornar informações sobre as operações do objeto e parâmetros necessários para chama-lo. Usando essas informações o cliente pode determinar como construir dinamicamente um pedido a um objeto.

Pode ser usado para checagem de tipos dinamicamente quando diminui as referências de um objeto a um tipo particular, para determinar quando um tipo de um objeto realmente combina com uma definição de um tipo local. Isso é importante quando um ORB recebe uma referência para um objeto que é gerenciado por outro ORB.

Pergunta: Quais são os passos para se desenvolver uma aplicação CORBA usando a IDL?

Resposta:

- 1.. Escreva uma especificação para cada objeto usando a IDL.
- 2.. Use o compilador (como o idl2java para gerar o código "client stub" e o "service skeleton"
- 3.. Implemente a interface do serviço.
- 4.. Escreva uma aplicação servidor para criar, ativar e registrar uma instância do serviço.
- 5.. Escreva a aplicação cliente.
- 6.. Use o compilador da linguagem de sua escolha para compilar a interface de serviço, aplicação servidor e aplicação cliente.
- 7.. Inicie a aplicação servidor.
- 8.. Rode a aplicação cliente.

Pergunta: Qual o papel da linguagem IDL (Interface Definition Language) na arquitetura CORBA?

Resposta:A IDL CORBA serve como a "cola" entre as aplicações cliente e servidor, e cada um dos dois ORBs envolvidos. Ela permite que componentes escritos em diferentes linguagens se comuniquem entre eles usando IIOP e o resto da arquitetura CORBA. Ela tem o propósito de definir as interfaces comuns a todos os objetos distribuídos que serão usados na aplicação baseada em CORBA. A sua compilação gera código na linguagem usada na implementação de cada aplicativo que usará os objetos cuja interface foi definida.

