

§ 13.6 Java 远程调用 (RMI)

- Java远程调用, RMI (Remote Method Invocation) 是一种分布式技术, 使用RMI可以让一个虚拟机 (JVM) 上的应用程序请求调用位于网络上另一处的JVM上的对象方法。习惯上称发出调用请求的虚拟机 (JVM) 为 (本地) 客户机, 称接受并执行请求的虚拟机 (JVM) 为 (远程) 服务器。

§ 13.6.1 远程对象及其代理

1. 远程对象：

驻留在（远程）服务器上的对象是客户要请求的对象，称作远程对象。

2. 代理与存根(Stub)：

代理的特点是它与远程对象实现了相同的接口。

存根（Stub）：一种特殊的字节码，并让这个存根产生的对象为作为远程对象的代理。

3. Remote接口

RMI为了标识一个对象是远程对象，即可以被客户请求的对象，要求远程对象必须实现java.rmi包中的Remote接口，也就是说只有实现该接口的类的实例才被RMI认为是一个远程对象。

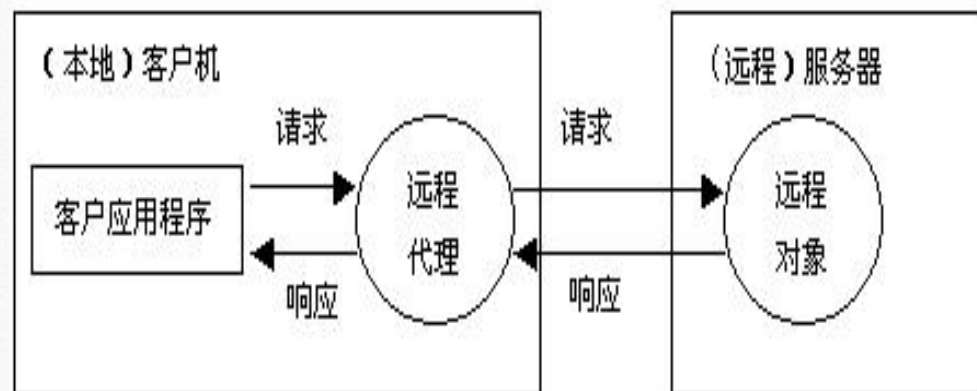


图 13.12 远程代理与远程对象

§ 13.6.2 RMI的设计细节

1. 扩展Remote接口

定义一个接口是java.rmi包中Remote的子接口，即扩展Remote接口。

RemoteSubject.java

2. 远程对象

创建远程对象的类必须要实现Remote接口，RMI使用Remote接口来标识远程对象，但是Remote中没有方法，因此创建远程对象的类需要实现Remote接口的一个子接口。

RemoteConcreteSubject.java

3. 存根（Stub）与代理：RMI使用rmic命令生成存根

RemoteConcreteSubject_Stub.class

§ 13.6.2 RMI的设计细节

4. 启动注册rmiregistry: 执行rmiregistry命令

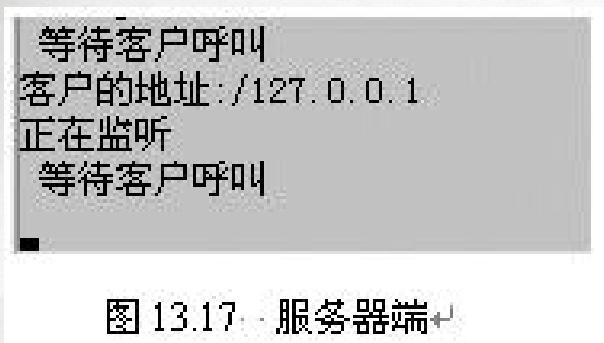
5. 启动远程对象服务: 远程服务器使用java.rmi包中的Naming类调用其类方法rebind(String name, Remote obj) 绑定一个远程对象到rmiregistry所管理的注册表中, 该方法的name参数是URL格式, obj参数是远程对象, 将来客户端的代理会通过name 找到远程对象obj。

BindRemoteObject.java

6. 运行客户端程序: 远程服务器启动远程对象服务后, 客户端就可以运行有关程序, 访问使用远程对象。 ClientApplication.java

§ 13.7 应用举例

- 查询服务器上数据库表的记录是最常见网络应用程序，本节利用套接字技术实现应用程序中对数据库的访问。应用程序只是利用套接字连接向服务器发送一个查询的条件，而服务器负责对数据库的查询，然后服务器再将查询的结果利用建立的套接字返回给客户端。
- 将例子7中的服务器端代码编译通过，并运行起来，如图13.17所示。 客户端输入车号或车名查询效果如图13.18



服务器端 Server.java , Target.java



客户端(Client.java)