

Excercise Sheet 5

Solution

Lecture Distributed Systems

Summer Term 2020

Exercise 1: Programming: Factories

You will find the solution to this task in the archive [105eFiles.zip](#)¹ on the lecture's web page. The `getAllAccounts()` method returns a serialized list of all accounts, but the actual contents of the list are remote references.

Exercise 2: Thread safety

The thread security of an RMI application can be achieved by using the keyword `synchronized` for the relevant method declarations in the server class:

Server class

```
public synchronized void add(int i) {
    number = number + i;
}

public synchronized int value() {
    return(number);
}
```

On the client side, a remote object cannot be locked (unless it contains a own remote methods specifically for this purpose). If the client would execute

```
Number num = (Number)Naming.lookup(...);
synchronized(num) {
    num.add(10);
    System.out.println(num.value());
}
```

only the local stub object would be locked, which would not result in mutual exclusion with the other clients.

Exercise 3: Programming: Client callback - progress bar

You will find the solution to this assignment in in the archive [105eFiles.zip](#)² on the lecture's web page. The mentioned problem when inheriting from the class `UnicastRemoteObject` is that the client does not terminate anymore. The reason is that a thread is created for a remote object that processes incoming remote calls. However, a Java program does not terminate until all threads have ended.

¹<http://www.bs.informatik.uni-siegen.de/web/wismueller/v1/vs/105eFiles.zip>

²<http://www.bs.informatik.uni-siegen.de/web/wismueller/v1/vs/105eFiles.zip>

Exercise 4: Name services

- a) IP addresses on the Internet are used for addressing hosts. To access a host, an IP address must be resolved into an Ethernet address, for example.
- b) Examples are ISBN numbers for books, ID numbers for software and hardware products, personnel numbers within a company, and Ethernet addresses (although some addresses are used to identify an entire machine and not just the Ethernet card).
- c) Both names may be position-independent, but the first gives less indication of the position of the specified unit. Position independence in this case means that the name of the unit is independent of its address. If one considers only the name, nothing can be said about the address of the corresponding unit.
- d) First of all, if aliases are only defined privately and not publicly, there is the risk that a user may erroneously but unnoticed reference a certain object that uses the alias. The alias could reference a (wrong) object in the namespace of another user. Furthermore, aliases can introduce cycles within a graph. For example, a name like `/users/fred` could in principle be made an alias of `/users`. A resolver could potentially end up in an endless loop while trying to resolve the name. One solution to this problem may be to set a limit on the number of aliases a resolver can test when trying to resolve the name.

Exercise 5: Name services

Name Service: Assignment of names to objects in order to find services. Physical addresses are unsuitable because they can change.

Binding: Assignment of a name to an object, resource, or physical address;

Examples: DNS (Host name, Mail exchanger), NIS (User + Name + LoginID)

Hierarchical naming services: Division of the namespace into zones, one server per zone exists, which knows the sub servers for its sub zones or the entries themselves at the bottom.

Iterative name resolution: Client queries the (sub)servers itself one after the other.

Recursive name resolution: Server requests subserver, this one its subserver, etc.

Exercise 6: Programmierung: JNDI

- a) JNDI (Java Naming and Directory Interface) provides methods with standard directory operations (reading, writing, deleting, adding attributes to objects, and searching for objects based on these attributes). In addition, it defines a unified interface for accessing various directory services.

Java applications provide two ways to work with names and directory services using JNDI. One is the traditional way - traditional in this context means to access actually existing directory services (file system, mail, etc.). The second possibility is to use the directory as a storage location and an organizational form for storing objects.
- b) You will find the solution to this problem in in the archive [105eFiles.zip](#)³ on the lecture's web page.
- c) You will find the solution to this problem in in the archive [105eFiles.zip](#)⁴ on the lecture's web page.

³<http://www.bs.informatik.uni-siegen.de/web/wismueller/v1/vs/105eFiles.zip>

⁴<http://www.bs.informatik.uni-siegen.de/web/wismueller/v1/vs/105eFiles.zip>