

BTC4201 / ICS4104: Distributed Systems

Assignment: Inter-process Communications in Distributed Environment

ICS4-Group A

Nyambok Julius/105865

David Adeola/91803

Muthoni Kahuko/111886

Andre Benhgi/94122

RMI can use any naming service but defines its own service, the RMI registry. The standard port for this service is 9090. The registry is handled by a standard JDK tool, `rmiregistry`

The steps that we took are the following:

- Interface definitions for the remote services
- Implementations of the remote services
- Stub (and Skeleton) files
- A server to host the remote services
- An RMI Naming service that allows clients to find the remote services
- A class file provider (an HTTP or FTP server)
- A client program that needs the remote services

SocketClient.java is the client class.

SocketServer.java is the server class.

ClientProtocol.java is the abstract connection interface.

SList.java is the base line for the connection for the log information.

ServerProtocol.java is the transaction class that shows the multiple users.

The multiple clients join the “chat-room” like interface where the server can direct the questions dynamically.

For example. A client in the chatroom called John can be asked by the server the stated questions stated which are:

1. What is your student number?
2. What is your admission number?
3. What is your personal code?
4. What is your faculty name?

References

Stepisnik, J. (2007). *Distributed Object-Oriented Architectures: Sockets, Java RMI and CORBA*.

Diplomica Verlag.

Stepisnik, J. (2007a). *Distributed Object-Oriented Architectures*. Diplomica Verlag.