|  |  |  |  |
| --- | --- | --- | --- |
| Task 0 | 使用Wireshark熟悉课程中各种协议 |  | 实验零 |
| Task 1 | 使用Java/C++实现Web服务器编程 |  | 实验一 |
| Task 2 | 使用Java/C++实现POP客户端编程 |  | 实验二 |
| Task 3 | 使用Java/C++/CORBA实现分布应用编程 |  | 实验三 |

参考资料：

1. HyperText Transfer Protocol, RFC 1945 RFC 2616
2. PostOffice Protocol Ver.3 RFC 1939
3. Java Network Programming and Distributed Computing
4. Practical Packet Analysis：sing Wireshark to Solve Real-World Network Problems
5. Instant Wireshark Starter

Task 0

利用Wireshark截取数据包，并对数据包进行解析。上交分析报告。

Task 1

完成简单的web服务器程序，上交程序和设计文档

Task 2

完成user、pass、retr、list等命令，上交程序和设计文档

Task 3

|  |
| --- |
| **Readings:**   * Reilly and Reilly, chapter 12 * JavaTM IDL Technology Documentation at <http://java.sun.com/j2se/1.5/docs/guide/idl/index.html>. These readings describe a variety of ways in which Java and CORBA interact, and give an introduction to using IDL and CORBA. |

### Background

Build a distributed to-do list using Java and CORBA. Various clients should be able to connect to a server managing various to-do list objects to query list items, add list items, delete list items, and clear to-do lists. The server should also have management functionality that allows member registration. Each to-do list item has a description, and a start time and an end time in which users must complete their items. Users should be able to add any number of items to their to-do lists.

### Tasks

For this programming exercise, we will be using JavaSoft's Java IDL, which is a CORBA ORB written completely in Java. The advantages of Java's IDL are that it is free and very portable. Its disadvantage is that its functionality is minimal. However, it supports static method invocations, and it has a naming service, which is all we need for this programming exercise. The Java IDL compiler (idlj) is packaged with the [latest version of Java](http://java.sun.com/j2se/1.4.2/download.html) (1.4.2 as of the time of this writing). You may find documentation for the idlj compiler [here](http://java.sun.com/j2se/1.4.2/docs/guide/rmi-iiop/toJavaPortableUG.html).

Your to-do list should implement the following functionality. Each piece of functionality outlined below must be accessible from the client using the CORBA naming service. Each user's to-do list must be a separate object that is managed using CORBA.

**User Registration**

New users may register. New users must supply a user name and a password. If the supplied user name is already in use by some other user, print an error message. Upon successful registration, print a message stating so. A new user registration should create a new to-do list object for the user.

**Adding Items**

Registered users may add items to their to-do list. Each item has a start time and an end time. Parameters should include the start and end times, and a label. The user should receive a message indicating a successful addition or an error if one occurs.

**Querying Items**

Registered users may look up all items in their to-do list in a given interval of time. The parameters should include the start and end times of the search interval. Return a list of the entries found in the specified time range. In the list, include the start time, end time, and label of each item.

**Deleting Items**

Registered users may delete items from their to-do list. The parameters should uniquely identify the entry to be deleted.

**Clearing Items**

Registered users may clear all items from their to-do list.

### Interface

Your client must be accessible via a simple console-based menu.

### Implementation

To complete this assignment, you will need to manage multiple CORBA objects using the CORBA naming service. For each user that successfully registers, you must create a new to-do list and add it to the CORBA naming service. To that end, you will need to write a to-do list creator that creates to-do lists and adds them to the CORBA naming system. The to-do list creator must be also be accessible using the CORBA naming service. Therefore, if there are n to-do list system users, CORBA must manage n+1 CORBA objects — one to-do list object for each user plus the creator object. You will need to perform the following tasks to complete this assignment.

* Write a to-do list creator object that creates to-do lists upon a successful user registration and adds the to-do list objects to the CORBA naming service.
* Casting the to-do list object to a generic Java object and generating the IDL description of the interfaces.
* Write a simple CORBA client that exercises the main methods of the to-do list and to-do list creator CORBA object.
* Deploying and running the system. You should run a test involving two or more agenda objects, keeping track of the agendas of different users. The client should use the CORBA naming service to find the appropriate agenda object.

### Submission

**Coding Style and Documentation**

You are expected to design classes in an object-oriented manner using good programming practices.

Your code must be written using [Java coding conventions](http://java.sun.com/docs/codeconv/). Your code must be well documented and contain a README file with instructions on compiling and running each program.

The files you should turn in are listed below.

* Submit all files necessary to run your to-do list.
* A README file.