CS332 - Project 2, Spring 2010

Assigned Thurs., Jan 28
Due date: 5:00pm, Friday, Feb. 12

Introduction

In the first project, you wrote a client that interacted with a server I provided. For the next project, you will be writing the server. The protocol is exactly the same for this project, so you should refer back to the Project 1 assignments for the relevant details. If you have thrown away the assignment, it is still available on the class assignments page.

The Server

The command line syntax for the server is

```
Project1Server -s <cookie> -p <port>
```

The server responds to correctly formatted client messages by printing the contents of the received message. When the server receives the BYE message, it adds that it has “received the final client message,” prints out this final message, and continues to run, waiting for additional clients that may wish to contact it.

Client modification

You should modify your client so that it can accept both dotted-quad IP addresses (the behavior for project 1) and domain names. for example, you should be able to type either

```
Project1Client -s 141.166.207.129: 5001 -f Ferd -l Berfle
```

or

```
Project1Client -s mathcs02:35001 -f Ferd -l Berfle
```

to connect to a running server on mathcs02. This doesn’t take a lot of work. The necessary changes are described in chapter 3 of Donahoo and Calvert. I’d suggest getting your server working first, then adding this change at the end. (However, depending on which examples you referred to in writing your code, it may already work this way.) If you didn’t finish Project 1 with a working client, let me know and I’ll give you my solution.

Notes

- Note that I am not providing a skeleton for you to start from. The code that handles command line arguments from the client should be a good enough example to get you going on the server argument handling. The example servers in the Donohoo and Calvert book should give you a starting point for your code.
• Obviously, since your client is using a stream based socket, your server should do the same.
• Remember the -lnsl flags for the link phase of compilation.
• You’ve got two different message types to handle for the server. This makes the code more complicated than the client.
• Don’t forget to close the client socket when you have finished the interaction with the client. If you forget, your server would eventually crash when it had too many sockets open.
• Don’t wait until the last minute to work on this.

Submission

Your tar file for submission should contain:

• your Project1Server.c
• your modified Project1Client.c
• your Makefile
• a README.txt file containing a writeup that gives a basic description of how your server works, written so that one of your classmates could read the description and correctly implement a solution to this problem, and including any known bugs in your programs, and the main problems you encountered in writing your server.

Give the tar archive a name that will be unique, like barnett_proj2.tar, and drop the tar archive in my Netfiles inbox (my user ID is lbarnett). You may resubmit before the deadline if you discover errors. Be sure to send me an email if you do this.