Matt Forbes

Assignment 5

March 4, 2011

1 Server Documentation

My server implements all elements declared in the RFA protocol. It passes both my tests and tests written by Dr. Nelson. The server relies on the functions defined in netlib.h, which along with netlib.c, should be included in the directory. There are a few ways that the code could be cleaned up, but I'd rather not break it in the process.

1.1 Overview

The bulk of the program is in the function handle_client, which is called by a forked process after a client connection is made. In this function, an array of all the open file descriptors is held. Rather than handing back actual file descriptors to the client, indexes to this array are passed. I'm not sure why I did this.

When a client makes a request, an action is performed based on the first character of the message. Currently, the request string is parsed character by character which is something that could be cleaned up.

1.2 Structures

Unfortunately, I didn't use any interesting data structures except for a linked list. It's main usage was in loading up the authorized hostnames; it made it easy to just read the .rfahosts line-by-line and throw it in to the list.

1.3 Tests

Unlike my client code, I didn't do much testing until the program was complete. After getting the Open procedure working and parsing down, the rest of the functionality just fell in to place. Besides running the 'try' test script, I played around with ncat to make sure files were being written and read correctly.

1.4 Possible Improvements

One thing that really bothers me about my code is how request strings are being parsed. It works just fine, but the same code is duplicated on multiple lines in multiple functions. My initial attempt was to use the strtok function from the string library, but that breaks when filenames have spaces. In hindsight, I should have used that method for all the requests except for Open, which is a special case.

2 Client Documentation