

CSE 4510/5241 HW1

Due 5pm, Feb 4, 2009

Submit server: `course=dc, project=hw1`

Design and implement a server (esftpd), a client (esftp), and the protocol for ESFTP (Extremely Simple File Transfer Protocol) that allows files to be transferred between the client and the server. The client allows the following commands:

- `get filename` (fetching a file from the server)
- `put filename` (depositing a file at the server)
- `quit`

You may assume the server can only serve one client at a time. Your implementation should display whether a file is transferred successfully and handle errors (for example, host not found/reachable, port is being used, file not found, not all bytes were successfully transferred...) gracefully. For the various stages of your server and client, print messages to the console to show what they are doing.

Since in real life, files are arbitrarily long, you can't (and don't want to) setup a buffer large enough to hold a file. Set your buffer size to be some small number of bytes (like 8 or 16 bytes) so that you can test your program on large files (larger than your buffer size). Usually, port numbers above 50000 are free to use. (In unix `/etc/services` has all the port numbers assigned by the OS.) [For initial testing, you might want to use `localhost` (127.0.0.1).]

Test scenarios should include:

1. putting and getting an ascii/text file
2. putting and getting a binary file
3. putting and getting non-existent files

A sample session might look like:

```
-- Server -----
shark% esftpd 55555
[Socket created]
[Socket bound]
[Listening at port 55555]
[Accepted a connection from dolphin]

[Request: put test1.txt]
[test1.txt opened]
[123 bytes received/written]
[test1.txt closed]
...
^C
shark%

-- Client -----
dolphin% esftp shark 55555
[shark is 163.118.22.70]
[Socket created]
[Connection established]

esftp> put test1.txt
[test1.txt opened]
[Sending data]
[test1.txt closed]
[Successful transfer of test1.txt (123 bytes)]

esftp> get test2.txt
...
esftp> quit
dolphin%
```

CSE 5241 students only

Allow multiple files to be transferred with one command:

- `get file1 file2 file3`
- `get *.txt`
- `get file.*`
- `get f*.*`
- `get *`

and similarly for `put`.

What to turn in:

1. Detailed description of your ESFTP protocol (in the README.txt file)
2. Compilation instructions (preferably makefile)
3. Source code
4. Sample session using `script` on unix (esftp-log.txt, esftpd-log.txt)

If you use code from books, remember to credit the source. I'll be testing your programs on the specified Linux machines (`dcom0 dcom1 dcom2 dcom3.cs.fit.edu`). If you are developing your program on other machines, you might want to test it on those Linux machines. In unix `od filename` (octal dump) is a tool you can use to "print" out a binary file in human readable form. I'll be using it to check if binary files are transferred successfully.