A Parallel FTP Client

Chuanqi Yan student ID: 12215531 CNetId: ycqstrong

Description

This project implements a parallel FTP **client** application using the socket API that can download files from the server end.

Behavior

When executing the program, you should firstly enter the arguments to indicate whether you are going to use the parallel downloading, the server's address, the client's username, password etc. The syntax of the command line should be:

Usage:
pftp [-s hostname] [-f file] [options] pftp -h | --help
pftp -v | --version

help: -h or --help
Prints a synopsis of the application usage and exits with return code 0.
-v or --version
Prints the name of the application, the version number (in this case the version has to
be 0.1), the author, and exits, returning 0.
[-f file] or [--file file]
Specifies the file to download.
[-s hostname] or [--server hostname]
Specifies the server to download the file from

Options:
[-p port] or [--port port]
Specifies the port to be used when contacting the server. (default value: 21).
[-n user] or [--username user]
Uses the username user when logging into the FTP server (default value: anonymous).
[-P password] or [--password password]
Uses the password password when logging into the FTP server (default value:user@localhost.localnet).
[-m mode] or [--mode mode]
Specifies the mode to be used for the transfer (ASCII or binary) (default value: binary).
[-l logfile] or [--log logfile]
Logs all the FTP commands exchanged with the server and the corresponding replies to file logfile. If the filename is "-" then the commands are printed to the standard output.

Parallel FTP Download
Usage: pftp [-t para-config_file] [options]
[-t para-config-file] or [--thread config-file]
Each line in the config-file specifies the login, password, hostname and absolute path to the file.

Note: pftp can use the --thread option simultaneously with the -l option. And pftp only supports binary model not ASCII mode.
For each line of the para-config file, the format should be: ftp://username:password@servername/file-path.
An example of the para-configuration file is as follows

ftp://cs23300:youcandoit@ftp1.cs.uchicago.edu/rfc959.pdf
ftp://socketprogramming:rocks@ftp2.cs.uchicago.edu/rfc959.pdf

Description of the code

The file analyze.c implements a parsing function to parse the arguments and does the initiation of the client and server.
The file connect.c implements the ftpconnect() function which can do the socket connection, command transmission, downloading the file and write the file to local file.
In the main function, if the client doesn't use the parallel download, it will simply call the ftpconnect() function to do the downloading task. Otherwise it will read the para-config_file and create n threads (n equals to the number of lines in the para-config_file). Each thread will call the ftpconnect function. In that function, the client will first login, and then send the "SIZE" command to get the size of the target file. The threads are going to separate the target file to several "subfiles". They will calculate the size of the subfile that they should download and will use REST to set the start position. Then it will download corresponding bytes to a temporary file. After all the threads are finished, the main function will combine the temp_files together to a final file.

Compile and execute

In the command line, after reaching the file of the project, enter "make" to compile the main.c file. Then execute the pftp file using the syntax of the command: ./pftp [-s hostname] [-f file] [options] or ./pftp -h | —help or ./pftp -v | —version or pftp [-t para-config_file] [options].

Output

If no errors happen, there will be no output in the client's end. If an error happens, the description of the error will be output to the stderr. If the logfile is "-", then the commands sent by user and the returning message will be output to stdout.