Unix System

my_ftp

Contact
b-psu-330@epitech.eu

Contents

Administrative details

- Your sources shall be turned-in on the `PSU_year_myftp` directory
  ex: PSU_2013_myftp for the 2013-2014 scolar year
- The whole project shall compile with a single Makefile.
- This project is to be carried out alone.

Subject

- The purpose of this project is to create an FTP client and server.
- Your are free for the choice of the protocol to use (you are not required to respect the RFCs).
- The network communication will be achieved with TCP sockets.

.1 Server

- Synopsis:

```
Usage : ./serveur port
```

  - `port` is the port number on which the server socket is listening.

- The server must be able to handle several clients at the same time by using a **fork**.

Indices If you are very comfortable with **select**, you can use it. Nevertheless, you will have the opportunity to use it during the my_irc project, so take advantage of this project to learn about a forking server.

If you use **select**, do things correctly:

- select for reading
- select for writing
- nonblocking server

If your server is badly designed because you wanted to use **select**, you will lose numerous points during the defense.

.2 Client

- Synopsis:

```
Usage : ./client host port
```

  - `host` is the name (or the IP address) of the computer where the server is hosted.
  - `port` is the port number on which the server is listening.

- The client must handle the following commands:
  - ls: list the current directory of the server.

  - cd: change the current directory of the server.
  - get __FILE__: copy the file __FILE__ from the server to the client.
  - put __FILE__: copy the file __FILE__ from the client to the server.
  - pwd: display the current directory of the server.
  - quit: close the connection and stop the program.

- The client must match the following requirements:
  - A specific prompt (to differentiate it from the shell).
  - Impossibility to browse the "file system" to a level lesser than the execution directory of the server (document root by default).
  - Display of the SUCCESS and ERROR (with explanatory message) for each request.

.3 Bonus

- lcd + lpwd + lls
- mget + mput
- rights management
- management of the user accounts and of the authentication (password)
- home directory different for each user account
- handle the "bin" and "asc" modes
- automatic completion for get
- automatic completion for put

.4 Super Bonus

- Compliance with the RFC

Forbidden functions

- recv
- send
- all the system calls that make a socket non-blocking.

Allowed functions

- the C library minus the forbidden functions