Network Assignment Documentation
Introduction to Socket Programming

Ahmed Tarek ElGamal (9)
Salma Abd Elaziz (28)
Mohamed Youssef (67)

November 19, 2017

Problem statement:

Use sockets to implement a simple web client that communicates with a web server using a restricted subset of HTTP.

Organization:

The code is organized in three code files and a header file, the three code files are server code, client code and functions code file, the functions file code contains some functions that is used two sides.

Server :

- Server accepts incoming connection requests form clients (GET or POST) and takes the appropriate action.
- Server keeps the connection open (persistence connection) - for dynamic amount of time depending on load of requests on server - waiting for new requests from the same client.

client:

- Client must read and parse a series of commands from input file.
- Client open a connection with server on specified host listening on the specified port number.

How to run code:

1. the file that has the client requests called read.txt, fill it with the request before running the code
2. run $make all in terminal, in the directory of the client and directory of the server.
3. run $./server (port-number) in the server directory.
4. run $./client in the client directory.

Important Functions:

- split (string s, string delimiter) : the function splits the given string s at the delimiter.
- get_client_request(vector¡string¿ v, int i) : function that separate the client requests read from file.
- get_server_request(string request) : the function puts the request read from the client file to the proper format of HTTP request.

- parse_request(string request): function that parse request coming from the client.
- send(int new_fd, std::vector¡std::string¿ curRequest): function that send files.
- recv(int new_fd, string fileName): function that recceives files.

Data Structures:

- strings
- vector¡strings¿
- some primitive data types