Task:

In this project we want to create a socks proxy server and client software based on concurrent programming and some network knowledge including building a tunnel which connects the client and server and constructing a client to enable browser.

The reason we want to build a proxy server is due to the Great Firewall of China (GFW). People in mainland of China can’t access some selected websites such as Youtube or Google. It’s the main instrument used by the government to achieve Internet censorship in China. So the proxy is very essential for users in mainland of China to bypass the Internet filtering and gain access to the websites like Facebook or Google.

System:

The purpose of the Socks Proxy is to build a tunnel that allowing traffic to bypass Internet filtering to access content otherwise blocked by governments or workplaces. The client and server are connected through TCP. Each time when the client receives a TCP request, it will open a port and the server has a corresponding port, so the client and server communicate at layer 5 (session layer). All the TCP connections are transmitted between the corresponding port.

The server has the socket that is bound to a specific port number, and it just waits, listening to the socket for a client to make a connection request.

The client should know the hostname (address) of the server, and the corresponding port the server is listening. To make a connection request, the client tries to rendezvous with the server on the server's address and port. The client also needs to identify itself to the server so it binds to a local port number that it will use during this connection. Once the server accepts the connection, the server gets a new socket bound to the same local port and has its remote endpoint set to the address and port of the client. The new sockets ensure it can continue to listen to the original socket for connection requests while tending to the need of connected client. While for the client, if the connection is accepted, a socket is created and it will use the socket to communicate with the server.

To make it simple, will use a simple authentication method in socks5 - No authentication. But the connection between client and server are encrypted with Blowfish encryption algorithm. So the key is required at the start of connection.

Difficulties:

With the help of some references, we got the basic idea of how socks works. But we need to dive into some codes to see how we can use Rust to realize those functions. Another problem for us maybe the lack of computer network knowledge. Actually, we have no idea how the socks5 works and what we can do to improve the efficiency of our product.

Must-have:

A proxy server application with tunnel to connect client and server.

A client that enable the browser or other app to connect to server.

Nice-to-have:

Maybe a user-friendly GUI or improve the efficiency by changing the protocol or adding more essential functions.