CITRINI Mathias

# Project 1: Web server

## Architecture of the project

Une image contenant texte, écran, noir, capture d’écran

Description générée automatiquement

|  |  |
| --- | --- |
| Makefile | makefile |
| README and .gitignore | Useful for GitHub dev |
| include/ | Contains Header |
| src/ | Contains source files (.c) |
| www/ | Contains website and all the assets (css, images) |

## Commands

### To compile the project, or clean it:

Command: make   
Command: make clean

### To start the server, use:

Command: ./server <port>

Replace <port> with your own port (ex: 8077)

## Webserver hub

I created a dummy website to test every requirement easily. This website is a basic one and use only one js script and one css file. It can display images, load video etc…

To access this website, please go to: <ip address>:<port>/index.html

Use Google Chrome to have better performance.

## How the server works?

Une image contenant diagramme

Description générée automatiquement

Une image contenant diagramme

Description générée automatiquement

## Request Analysis

## Difficulties

The first difficulty was when I had to parse the HTTP request. Sometimes, I had core dumped so I had to choose the best size for each field and optimize my “unparseHTTPRequest” function.

The second difficulty was to understand how to send binary files (Images, Audio …). In fact, I took some time to understand how binary files work and how to read it. I learned the best way was to use void\* instead of char\*, and so, memcpy instead of strcat. After long hours of pain, I finally managed to get the proper size of the file, read it with fread and copy the binary data in a string to send it to the client.

To summarize, memory problem and binary data were my biggest difficulties in this project. All other things were easy to do and didn’t take me too much time.