

# Project 1

CS 118

Siddhartha Bose 704579512

David Chen 904622289

### High Level Description of Implementation:

1. Parse HTTP request to get file name
2. Check directory to see if file exists, if not return 404
3. Parse file name to get desired extension, and set type in header
4. Open file and check file length
5. Send header with appropriate information
6. Read file into buffer
7. Send buffer over socket
8. For files with unidentified extensions, but do exist in the server directory, send across the socket and be downloaded.

### Difficulties Faced:

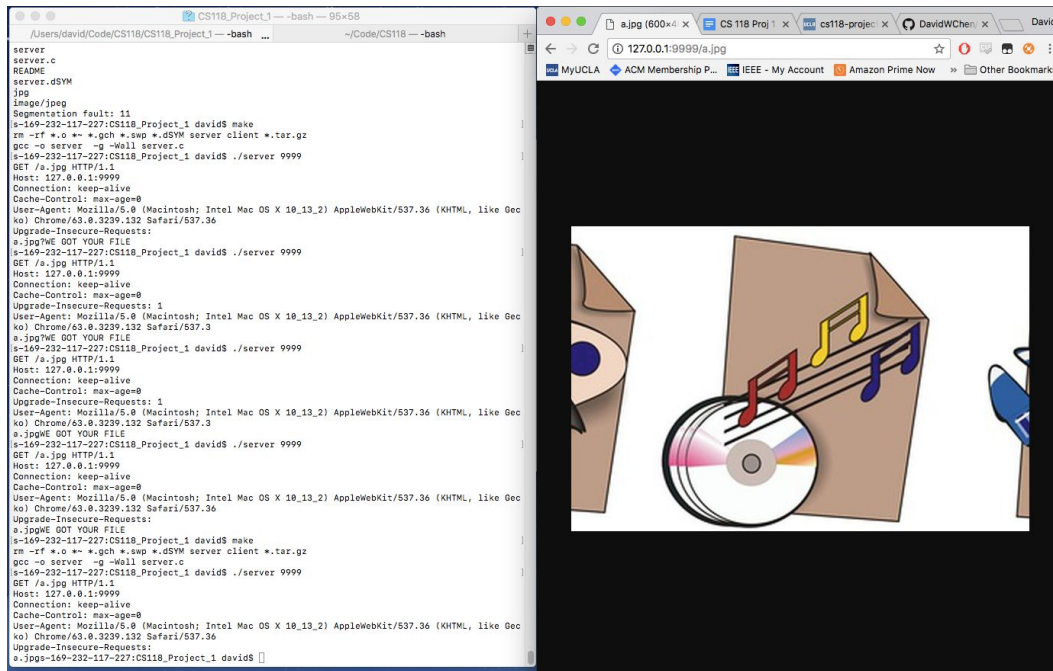
1. Dealing with extraordinarily long files. In our original implementation we appended every file to the end of the buffer and as such ran into segmentation faults. To fix this we decided to stop dynamically allocating memory onto the heap and hope the file fits and instead just read in the file into a buffer and sent it as a second message after sending the original header.
2. Dealing with spaces was a hassle until we understood that in the backend the spaces were converted into "%20". We then proceeded to use a string replacement algorithm for C (which we found on <https://www.geeksforgeeks.org/c-program-replace-word-text-another-given-word/>) and used it to replace any instances of a "space" with an actual space character.
3. In general using C was difficult since a lot of the functionality we are used to with generally higher level languages, even C++ was not available to us. As such we often suffered from segfaults and memory leaks. We needed to find creative and often odd methods to get around these roadblocks with certain functions to manipulate the data just like we wanted them to be.

### How to Compile:

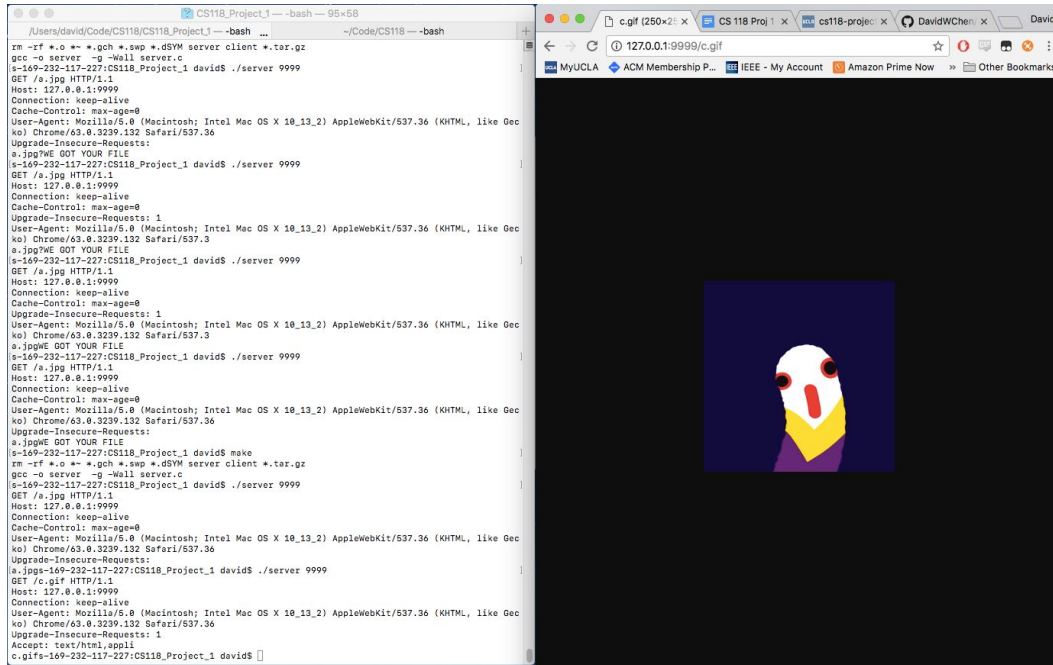
1. We edited the original makefile from the template so that once "make" is typed into the command line it runs make clean and make all.
2. From the standpoint of the client, the best way to run it locally is to put the local IP into the URL followed by /<filename>
3. This will then either display the chosen file type, or download it if it isn't one of the predefined extensions.

## Sample Outputs:

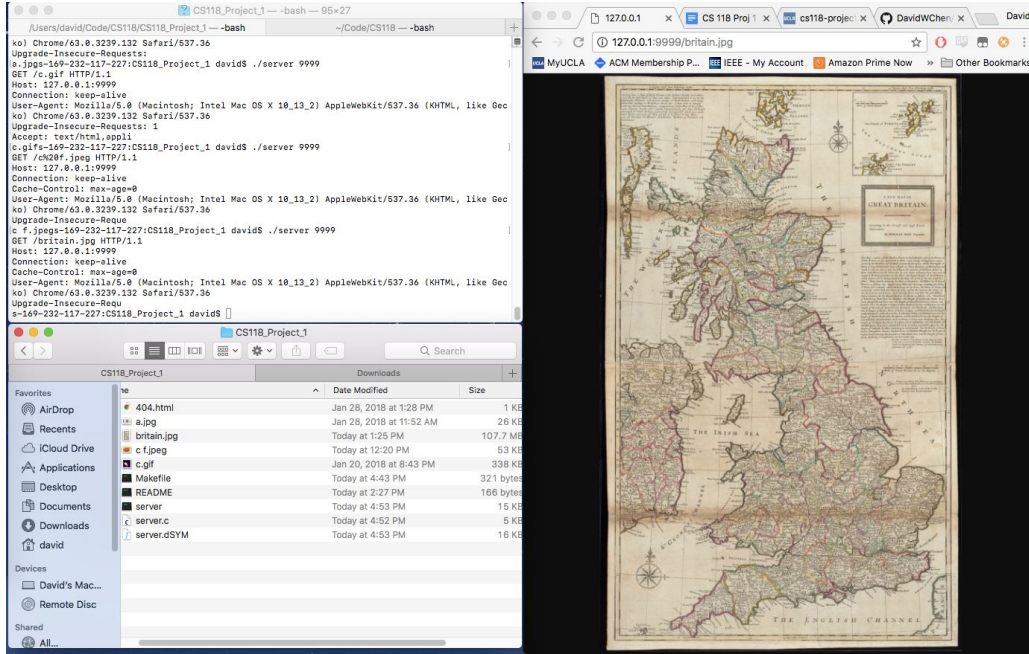
- Small binary file/Image
  - Jpeg



- gif



- Large binary file (up to 100 MiB)



- HTML file/404 response

