Design Document
Billy Kwong
Bqkwong
CSE 130 Fall19

ASGN1 Design

Goal

The goal of this program is to provide the simplest features of an httpserver(web server).

Assumptions

I'm assuming that the server will run on Unix environment and user will generate header as required by server else it will generate the error. In addition, it will only support the GET and PUT request from user and present working directory of user will serve as the storage of resources.

Design

The general approach I'm taking is to make the server listen to a queue on clients one by one (as it is a single threaded server) it will read the request header of client and perform some operation (PUT or GET) on the basis of that request and provide the response header. It will generate an appropriate error if the request gets failed or server crashes.

Pseudocode

define BUFFER_SIZE 4096
define DEFAULT_PORT 80

Procedure httpserver

Int port = DEFAULT_PORT;
If argc == 3
    Port <- toInteger(argv[2]);

server_sockfd <- 0
client_sockfd <- 0
Define structure sockaddr_in server_addr;
Define structure sockaddr_in client_addr;

Create passive socket for the server
Create an address structure containing server IP addr and port, then
Bind the server_sockfd with this address

Create connection queue and wait for clients

While true
    Accept a connection, blocks until connection from client is establish
    Will return a brand new descriptor for comm with this single connection

```
        If client_sockfd == -1
                Display_error “HTTP/1.1 500 INTERNAL SERVER ERROR\r\n”
                Continue;

        Else
                Count <- 0;
                Read from sockfd
                Define buf[BUFFER_SIZE];
                Define rv;
                While rv <- receive data from client
                        Tokenize the string and push them in vector
        If GET request
                If file doesn’t exist
                        Display_error “HTTP/1.1 404 NOT FOUND\r\n\”
                Else if exists but not accessible
                        Display_error “HTTP/1.1 403 FORBIDDEN\r\n”
                Else
                        Fd <- open file descriptor
                        While data in file
                                Send data to client_sockfd
                        Close fd;
                        Display_error “HTTP/1.1 200 OK\r\n”
        Else if PUT request
                Fd <- open file descriptor
                If fd == -1
                        Display_error “HTTP/1.1 403 FORBIDDEN\r\n”
                Else
                        If count <- content length
                        Close fd;
                                Display_error “HTTP/1.1 201 CREATED\r\n”

        Else
                display_error=”HTTP/1.1 400 BAD REQUEST\r\n”

        close  client_sockfd
return
```