

# Network Programming Phase2

## httpserver

---

*Learn diary*

Liu Peng 280600

### **To fulfill the requirements in the assignment:**

1. Server should serve HTTP GET/PUT request from multiple clients
2. Test interoperability with other implementations
3. Server should work also with standard web browsers
4. Server should support index method to list every file name on the server directory

### **Problems and solutions during programming:**

#### **1. How to differentiate the port number used for the service?**

To run the program, the second argument is used to define the listening port, but if the second argument is left empty, the default port number is predefined as 8888.

#### **2. How to keep the server is always running?**

To keep the server running even if I logged off the system, I created a daemon process that running in the background, listening to the defined port number. The daemon process ignored the signals except for the terminate and hang up signal, which makes the server running safely in the background.

To decide the process id, I also save the process id in a standalone file, which makes it easier for the administrator to see what pid is and kill the process when needed.

#### **3. How to serve multiple clients?**

To support several clients at the same time, the server should keep listening to the socket, so one idea came to my mind, to fork a child process for each incoming request, and the parent process keeps listening to the socket.

The bad effect of this method is that the fork action is quite heavy for the CPU, and it can waste more memory. When the amount of user is too large, this method should not be applied.

#### **4. How to implement the index function?**

The server should also support the index method to list all files on the server. To implement the function, I used `readdir()` function, which returns the filename of the given directory, I save all the file names in the buffer, calculate the string length and send back the response message with the standard HTTP format.

## 5. When testing the server in my chrome browser, the my server encountered a problem and did not response.

When I look into this problem I realize that there is a lam header field in my server that is assumed will appear for sure. However, the standard web browsers do not support this header, so I modified my code so that it can judge if there is a lam header in the request, if not ignore this part of header. Then I tested again with my chrome browser, problem solved.

## 6. How to keep a log file to record every event that happened to the server?

To keep a log file, I write a log\_message() function which can save the given message in certain format with time stamp to the log file. So when the program need to record something , just call this function. The open method is append so that the log file can keep all records.

### Function test:

features work	features don't work
Support GET request, can deliver the file on local disk to the requesting client	Only work for plain text
Support PUT request, can accept file from the client and create and save with the given file name on local disk	The time out function has not been implemented yet.
lam header works -support to identify client by lam header and save it in the log file	
Tested it's working for standard URLs and other web browsers	
Large file test successfully -support put and get request to transfer a 520KB file successfully	
Support index function -can list all the file names under the root directory and checked with my Chrome browser	
valgrind test passed -no leaked memory -no errors in memory operation	
Program uploaded in the netlab server and keeps running 2 days without problem	
Keep a log file that records all events that happened on the server	
Keep a daemon.lock file to indicate the running service process id -make it easier for administrator to manage the service	