

Assignment 5

Basic practice questions

1. Create a command line download program which takes URL as argument and downloads the file to current directory. You can send all requests to proxy so that you do not have to convert hostnames to IP addresses and so that files hosted over Internet can be downloaded by your program.

Assignment Questions

1. Create a web server which accepts port number and document root as command line arguments. The web server should parse request sent from browser and send appropriate file to client or '404 Document Not Found error', if file does not exist. Web server should also send the appropriate 'Content-Type: text/html' header based on whether file is '.txt', '.htm' or '.html'. For other files you can send 'Content-Type: application/octet-stream'.

Assuming after compiling your program we get file 'web_server' then, we should be able to run 'web_server' as

```
./web_server 9999 /home/example/public_html
```

After this whenever someone opens `http://<IP>:9999/index.html` in browser he/she should see contents of file `/home/example/public_html/index.html`. Similarly if someone tries to open `http://<IP>:9999/folder1/index.html` then he/she should see file `/home/example/public_html/folder1/index.html`. Also if file extension is '.gz', '.pdf', etc. then ensure using 'Content-Type' explained above that browser does not display funny characters on screen and rather provides "Save As" or "Download" option.

2. If the request contains GET or POST variables then print them on screen. For example if we type in browser 'http://<IP>:9999/index.html?a=1&b=2' then send 'index.html' file to browser as before and on console print 'GET variable a=1' and 'GET variable b=2'.

When values are submitted using POST from browser, then below HTTP request headers in HTTP request there is '\r\n\r\n' sequence then post values are present in

```
<variable>=<value>[&<variable>=<value>]
```

format. Or in simpler terms there is a blank line separating HTTP headers and variables in POST request.

Note that all variable names and values are URL encoded (or percent encoded) so that they do not contain '&', ' ', '\r', '\n', '=', etc. which can cause problem. Hence you have to separate different variables by '&', then variable from its value by '='. Then URL decode both variable and value and then print them on console.

For POST variables also print in format 'POST variable a=1' and 'POST variable b=2'

Advanced practice questions

1. Create a command line download program which accepts URL as argument and download the file in current directory. You can open parallel sessions to server and use 'Content-Range:' Header to download file in small portions.

Refer to <http://www.w3.org/Protocols/rfc2616/rfc2616-sec14.html> point 14.16 to understand how to use Content-Range. Optionally you can use a download manager and capture packets using Wireshark and see who the download manager is using HEAD requests and 'Content-Range' headers to download files in multiple parallel streams.

2. Modify above program so that it reads 'http_proxy' environment variable and downloads files through proxy server. The download manager can also accept number of parallel sessions to open as second command line argument.