Assignment 2

HTTP Client and Server
Due: February 3, 2015

Write a TCP concurrent Hyper Text Transfer Protocol (HTTP) client and server, namely, *MyBrowser* and *MyHTTP*.

Your client and server should support the two basic HTTP commands, GET and PUT. The GET and PUT commands are for downloading/replacing document(s) from/to server. Read the RFC for HTTP (RFC 2616) for details of commands. You can also look up textbooks and other internet sources in addition.

HTTP Client (*MyBrowser*):

- *MyBrowser* is executed in the client machine. If successful, it waits on "*MyBrowser*>" prompt for command.
- Three command-line operators are supported: GET, PUT and QUIT. The GET and PUT commands are for downloading/replacing document(s) from/to server. After every GET or PUT command, the prompt returns to *MyBrowser*>. On entering QUIT, the program exits from *MyBrowser*> prompt.
- The GET command implements a simple version of the corresponding GET command in HTTP:
    - Opens a TCP connection with the http server.
    - Download/retrieve document from the http server.
    - Format: *MyBrowser*> GET {URL}
    - URL specifies the url of a web document. The default port number is 80. Example:
        i. Example: *MyBrowser*> GET http://cse.iitkgp.ac.in/~agupta/compsyslab/index.html
        ii. *MyBrowser*> GET http://10.98.78.2/docs/a1.pdf:8080
    - On receiving the document from the http server, the client opens the document by using appropriate applications. You have to support the following document types and applications: *Adobe Acrobat* for PDF files; *Mozilla* for HTML page, *gedit* for all others.
- The PUT command implements a simple version of the corresponding PUT command in HTTP and is used to upload/replace document to the server.
    - Opens a TCP connection with the http server.
    - Uploads/replaces a document at the http server.
    - Format: *MyBrowser*> PUT {URL} <filename>
    - URL specifies the url of a web document. The default port number is 80. Example:
        i. *MyBrowser*> PUT http://cse.iitkgp.ac.in/~agupta/compsyslab assignment.pdf
        ii. *MyBrowser*> PUT http://10.98.78.2/docs:8080 biodata.txt

HTTP Server (*MyHTTP*):

MyHTTP receives the "GET" or "PUT" requests from *MyBrowser*. It then parse the input stream, extracts the fields, and then sends back appropriate response. In case of error, appropriate error numbers and messages are returned to the client.

MyHTTP maintains an Access Log (AccessLog.txt) which records every client accesses. Format of every record/line of the AccessLog.txt: <Date(ddmmyy)>:<Time(hhmmss)>:<Client IP>:<Client Port>:<GET/PUT>:<URL>

All messages should be in proper HTTP format. Following header fields should be supported for this assignments. Other fields are to be ignored (note that ignored does not mean they will not occur in any message; just that they need not be handled).

- General Header: Cache-Control, Connection, Host
- Request Header: If-Modified-Since
- Response Header: Date, Location, Retry-after
  Status Codes: 200, 201, 403, 404, 408, 503, 504
- Entity Header: Content-language, Content-length, Content-type, Content-encoding, Last modified

For the fields mentioned above, only the following values (in case multiple values are possible) have to be handled now, the rest can be ignored (note that ignored does not mean they will not occur in any message; just that they need not be handled).

- Cache-Control: no-cache
- Connection: close
- Content-Language: en, en-us, en-gb
- Content-type: text/plain, text/html, application/pdf, image/jpeg
- Content-encoding: gzip

The client should print an appropriate message for the status codes defined above. For any other status codes, which must be some other error, the client should print a generic message such as "Unknown error" along with the code received.

You should submit two C files, the *MyBrowser.c* and *MyHTTP.c*.