Project - 2
CSC 4304 – Systems Programming
Due: December 3rd @11:59pm

1. Preparation

Before beginning your work, please read the following carefully:

- Sections 15.5 - 15.6 from Stevens
- Lecture slides on Network Programming and Concurrent Programming
- RFC1945 (http://ftp.ics.uci.edu/pub/ietf/http/rfc1945.html)

2. Programming Task: Implement a Simple Web Server

The objective of this project is to implement a simple web server "myhttpd" in C.

SYNOPSIS: **myhttpd** [−**dh**] [−**l** *file*] [−**p** *port*] [**-r** *dir*]

DESCRIPTION: **myhttpd** is a simple web server. It binds to a given port on the given address and waits for incoming HTTP/1.0 requests. It serves content from the given directory. That is, any requests for documents is resolved relative to this directory (the document root – by default, the directory where the server is running).

2.1 OPTIONS:

−**d** Enter debugging mode. That is, do not daemonize, only accept one connection at a time and enable logging to stdout. *(without this option, the web server should run as a daemon process in the background).*

−**h** Print a usage summary with all options and exit.

−**l** ***file*** Log all requests to the given file. See LOGGING for details.

−**p** ***port*** Listen on the given port. If not provided, **myhttpd** will listen on port 8080.

-r ***dir*** set the root directory for the http server to *dir***.**

2.2 DETAILS:

myhttpd speaks a simplified version of HTTP/1.0: it responds to GET, POST, and HEAD requests according to RFC1945. When a connection is made, **myhttpd** will respond with the appropriate HTTP/1.0 status code and the following headers:

Date	: The current timestamp in GMT.
Server	: A string identifying this server and version.
Last-Modified	: The timestamp in GMT of the file's last modification date.
Content-Type	: text/html or x-blowfish
Content-Length	: The size in bytes of the data returned.

If the request type was a GET, then it will subsequently return the data of the requested file. After serving the request, the connection is terminated.

POST requests will be used to send additional information to the web server. Please read: http://www.jmarshall.com/easy/http/#postmethod for more information on POST requests.

If the request was for a directory and the directory does not contain a file named "index.html", then **myhttpd** will generate a directory index, listing the contents of the directory in alphanumeric order. Files starting with a "." are ignored.

If the request begins with a '~', then the following string up to the first slash is translated into that user's **myhttpd** directory (ie */ classes/cs4304/<user>/myhttpd/*).

2.3 LOGGING:
By default, **myhttpd** does not do any logging. If explicitly enabled via the **−l** flag, **myhttpd** will log every request in a slight variation of Apache's so-called "common" format: '%a %t "%r" %>s %b'. That is, it will log:

%a : The remote IP address.
%t : The time the request was received (in GMT).
%r : The (quoted) first line of the request.
%>s : The status of the request.
%b : Size of the response in bytes. i.e, "Content-Length".

(check http://httpd.apache.org/docs/1.3/logs.html for more info on Apache logs.)

All lines will be appended to the given file unless **−d** was given, in which case all lines will be printed to stdout.

2.4 MULTITHREADING:
Each incoming request should be served via a new thread created by the main process.

2.5 SUPPORT FOR cgi-bin:
If the request is to a file with extention ".cgi" in the $ROOTDIR/cgi-bin/ directory, then the server should execute this file (probably a script or binary) and return the output of this program to the client. You should be using **pipes** to communicate the output of this program back to your serving thread.

For more information on CGI, please read: http://www.parkansky.com/tutorials/bdlogcgi.htm

3. How to Submit

After you finish implementation of your project, you can submit your source code as a single C file (renamed to **prj2_yourlastname**) by emailing to thanks@csc.lsu.edu and cc'ing to kosar@cct.lsu.edu. Please make sure you receive a message from the TA acknowledging the receipt of your submission, otherwise assume it is not submitted successfully.

4. Requirements

No report is required for this project. However, you are expected to submit the program with clear and sufficient comments and explanations. Presentation will be counted as a part of grading. Also make sure you perform all necessary error checks. All the programs should be submitted by **December 3rd @11:59pm.**