PROJECT: PHASE#1

CS 4/55231 INTERNET ENGINEERING

Spring 2012, Department Computer Science, Kent State University
Instructor: Javed I. Khan

Due Date: Feb 23 (3 weeks)

Project Objective:
The objective of the project in this class is to give you a practical experience of developing a mini Web Browser **MiniC** and a mini Web Server **MiniS**. These MiniBS together will support HTML document sharing over Internet. Eventually we will also implement several state-of-the-art technologies available in most modern web Server/Browsers. These include dynamic server side include, remote publishing, server-push client-pull, Forms and CGI support.

Phases:
To make our goal manageable, we will do the implementation in three phases distributed over the next few weeks. I will explain the technologies in the class. These parallel project phases will guide you through the practical part of these concepts. User interface can be text based. However, any graphical user interface will earn you 10% bonus.

Groups:
The projects can be performed as individual or as group project with up to 2 members. Each member of the group should assume a very specific responsibility. But every one will be fully responsible to understand and answer the entire project. All reports however should be individual. In the last page of each project report, you (and each of your friends) will include a confidential **Group Member Evaluation Table** indicating the contribution of the members (who did what) and a rough percentage workload distribution. This Table will remain confidential and will not be disclosed to your group members. Your total score will be a combination of the scores of the group project and your report.

Server Client Pairing:
Each group will be identified by a group number. And each group will implement both the MiniC and MiniS. You should create a directory called "IN2012S", and under it two subdirectories "IN2012S/MINIC" and "IN20012S/MINIS". In these you should keep all the MINIC and MINIS files. The IN2007 directory should contain an HTML file "INDEX.HTML", with pointers to a "README" file. During submission you will be required to zip the entire IN2012S directory and mail it to Grading/TA account:

projectsforjaved@gmail.com

Server Client & Submission:
The description of the first phase for both server and client is given below. At the end of each problem, I have specified the submission policy. I have assumed that a WILDFIRE group is doing this project. While naming your files substitute WILDFIRE by your actual group NAME. You should develop the system in such a way that it should run on unix/linux machines in our department network. If no port address is specified in MiniC URL or in MiniS initialization, the server should use 3341 as the default port.

Demonstration:
At the end you will be required to demonstrate the project. Projects successfully completed in time will be eligible for class demonstration and will receive automatic 10% bonus.

Phase#1: MINIC CLIENT

1. (1000 points) Write a routine Read_URL() for pursing URLs. For example, given an URL of the form:

   HTTP://www.cs.kent.edu:8000/~javed/class-IN12S/sample.html

This routine should be able to isolate the protocol scheme (HTTP), domain name of the server (www.cs.kent.edu), port number (8000) and the target name. Write a small main program minic-WILDFIREPROG1 (minic-WILDFIREPROG1.c) which accepts an URL from keyboard, use the routine Read_URL() and prints out the components. The port number is optional. If nothing is specified use the default.

Submission: Copy this version of your program into minic-WILDFIREPROG1.c and minic-WILDFIREPROG1.h files. Make it a part of your master makefile. Name of the executable should be minic-WILDFIREPROG1.

2. (2000 points) Modify the uptime client and Implement a MiniC which given an URL fetches the document from MiniS. If we type in "% minic-WILDFIREPROG1 -s URL" It should follow these steps:

A. Parse the URL and display the components of the URLs. Obtain and show the IP string address of the server.

B. Connect to the specified MiniC server. If any error is encountered, such as server not responding, IP address is wrong it should generate appropriate error message and return.

C. After setting up the connection, request a document by sending the following ASCII string "GET *targetname* MINIC/12S-WILDFIREPROG.1.0". *targetname* is the requested document.

D. Wait for MiniS to return a response string. If the document is present, the MiniS should return the following message with the above ASCII string.

   ourHTTP/1.0 200 Document Follows
   Content-type: *documenttype*
   Content_length: *length*

E. Then as a separate message it should receive the file from the MiniS, in a subdirectory called Cache. MiniS should then notify us on the screen that:
   "Document *targetname* of *length* bytes of type *documenttype* has been retrieved and saved in directory Cache"

F. If there is an error, the server will send the following message.

   ourHTTP/1.0 403 Not Found
   Content-type: *documenttype*
   Content_length: 0

MiniS should then notify us on the screen that:
"Document sample.html not found."

Submission: Copy this version of your program into minic-WILDFIREPROG2.c and minic-WILDFIREPROG2.h files. Make it a part of your master makefile. Name of the executable should be minic-WILDFIREPROG2.

3. (1000 points) The objective of the final part of this phase is to obtain a multimedia document. Multimedia documents generally have reference to other images and multimedia components in it. A Browser looks into the content of an HTML document and automatically retrieves any multimedia content embedded in it. Each content is fetched as a completely separate request. Now Modify MiniC so that if user starts it with flag with "%minic-WILDFIREPROG2 -full URL" it should do the following additional steps after obtaining the URL. (if we type in "%nimic-WILDFIREPROG2 -s URL" it should be back to simple mode, i.e. no multimedia).

A. It should look for any HTML IMG tag such as:
   <IMG src="http://www.mcs.kent.edu:8000/flowers.jpg>
B. Initiate communication with the specified MiniS to obtain the document using the procedures you have implemented in problem-2.

Submission: Copy this version of your program into minic-WILDFIREPROG3.c and minic-WILDFIREPROG3.h files. Make it a part of your master makefile. Name of the executable should be minic-WILDFIREPROG3.

Look at the end for final submission instructions.

Phase#1: MINIS SERVER

1. (1000 points) The first part of your assignment is to modify the uptime server into a concurrent Server. It should be able to accept multiple service requests simultaneously from several clients and serve them concurrently by initiating children processes. It should be able accept connection from a client and display the 32 bit HEX IP address, the Dotted Decimal IP address, and the PORT number of the client. (Hint: develop a routine called Client-ID() which should take the socket address as argument and return the above three information pieces. Call it from within the uptimeserver after it accepts a connection).

Submission: Copy this version of your program into minis-WILDFIREPROG1.c and minis-WILDFIREPROG1.h files. Make it a part of your master makefile. Name of the executable should be minis-WILDFIREPROG1.

2. (2000 points) The second part of your assignment is to implement a MiniS which now should be able to find and serve a document requested by a MiniC. The server should have a "home" directory. And the path of the requested documents should be interpreted relative to this home as root directory.

A. Accept a connection. Check for error. If OK, receive the message from the MiniS. Display the IP string address of the client, and the message.

B. Purse the message (look into MINIC specification for the message format). and separate the METHOD, URL and Client Information part from it. Display all three, separately.

C. If the METHOD is GET, then look for the document under the root directory (or subdirectory). If the document is not there, or any other error occurs return the following message in one ASCII string to the MiniC and close connection.

    ourHTTP/1.0 403 Not Found
    Content-type: *documenttype*
    Content_length: 0

D. If the document is available, obtain the length of the file. Look into the file name extension of the document. Use the file FILETYPE.txt in your course directory to obtain the *documenttype* of each document. And prepare a return message of the form:

    ourHTTP/1.0 200 Document Follows
    Content-type: *documenttype*
    Content_length: *length*

E. Send the response message to the MiniC.
F. Then send the actual document in a second message.
G. Now log the transaction in a log file called MINIS-WILDFIREPROG1.log. Append the IP string address, time, METHOD document name, and service status (200 or 403).
H. Close connection.

Submission: Copy this version of your program into minis-WILDFIREPROG2.c and minis-WILDFIREPROG2.h files. Make it a part of your master makefile. Name of the executable should be minis-WILDFIREPROG2.

3. (1000 points) The objective of the final part of this phase is to add a simple server side include mechanism in MiniS. If the document type is HTML, the server should look into the content and look for a echo comment tag inside it with a name of an UNIX environment variable. If there is one, it should substitute the tag with the value of the variable. Do the following:

a) Look for a tag of the form <!-#echo var="TIME"-> (TIME can be anything else).
b) If the doucment contains this tag, then read the unix environment variable TIME. If it is NULL then do nothing.
c) If it has a value, then substitute the entire comment with the value in the document (you should use a copy, and should not replace the original document), and send this modified document to the MiniS.

Submission: Copy this version of your program into minis-WILDFIREPROG3.c and minis-WILDFIREPROG3.h files. Make it a part of your master makefile. Name of the executable should be minic-WILDFIREPROG3.

SUBMISSION PROCEDURE

Each group should submit the project codes once. But, the reports should be individual. The project directory should be mailed as a ZIP file by at least one of you. The each of you should individually send your report.

For the Project (50% points):

1. Create an "README1.HTML" file in your "IN2012S" directory. This file should explain:

    1.1. A list of files included in this submission.
    1.2. A clear description how to compile,
    1.3. A clear description that how to use (start, terminate and run) your program.
    1.4. Include any special instruction to the grader if it has unusual interface.

2. Create a ZIP file called WILDFIREPROJ1.ZIP with the "IN2012S" including the README1.HTML file and its sub-directories.

Individual Report (35% points):

3. Each of you should now prepare one document called as "REPORT1-*YOURLASTNAME*.DOC" (It can be HTML, MSWord, or TXT file) as a phase report. For each part of the problem, It should describe:

    1.1. How you have implemented your part of the problem at logical level.
    1.2. If you are using any new API subroutine, you should explain why you are using that.
    1.3. You should also explain the design of your modules. Explain all the user defined variable, key data structure, etc. special files that you have defined to get the job done.
    1.4. One page **Group Member Evaluation Table, with three columns and a row for each member. In the columns describe (a) what part of the project a member has done, (b) your evaluation about the contribution of the member in percentage.**

Code Documentation (15% points):

4. Add documentation In the codes:

    4.1. At the top of each piece of your source code, include the following:

```
/*****************************************************************************
GROUP: xxxx
MEMBERS: 1. Xxx, 2.xxxx, 3.xxxxx etc.
DATE:
CS 4/55231 INTERNETENGINEERING 2012 SPRING
INSTRUCTOR: Javed Khan
*****************************************************************************/
```

    4.2. In each subroutine inside your source code include the tag:

```
/*****************************************************************************
GROUP NAME: xxxx
MODULE DEVELOPER 1. Xxx, 2.xxxx, 3.xxxxx etc.
MODULE DESCRIPTION: briefly say what it does? how it does?
*****************************************************************************/
```

    4.3. Also include ample comments inside your code explaining the program.

Final Mailing:

5. You now need to mail all of them to the course account:

    5.1. Mail the Project ZIP file to TA account projectsforjaved@gmail.com by email with subject fields "IN2012S WILDFIRE PROJECT1"

5.2. Mail the Report it to TA account by email with subject fields "IN2012F WILDFIRE REPORT1 [your last name]".

Check thoroughly before you submit. Keep a copy of all the files including shar.project1 in your directory. Do not modify them afterward. If need arises, TA may want to check these files. Any modification afterward (reflected in the file date) will result in late submission penalty.

Grading:
See notes to grader in the website.

Cheating and Copy:

If a copy is caught, all involved submissions (original as well as the copies) will be penalized. So it is your responsibility to guard your work. Secure the read/write access of your directories. Any copy will result in ZERO grade for the assignment for both parties. Only exception is when you report the theft of your work in advance.