Homework 4

Due: Tuesday 10/27 at noon

Complete the following, which asks you to continue lab 6, creating a simple web browser. Submit your work via handin as assignment hwk4. You only need to submit the last part since it includes the others.

For testing your browser, you want simple hand-written web pages. I suggest our course webpages: http://courses.knox.edu/cs226/main.html (the right frame) and http://courses.knox.edu/cs226/menu.html (the left one, with the menu).

1. (5 points) Write a C program that takes a URL beginning with “http://” from the command line and prints the host name and file name. The file name should begin with a slash.

2. (5 points) Expand the program from the previous part so that it sends a HTTP request to port 80 on that host. You can use SetupTCPClientSocket from TCPClientUtility.c to perform the address lookup and open the connection; use “http” as the name of the service. (This code is discussed in the lab handout and in section 3.2.1 of the text.) The entire response should be printed to the screen. Note that you may need more than one call to recv in order to get the entire response. (Keep reading from the socket as long as you get bytes back; use the return value of recv.)

3. (5 points) Expand the program from the previous part so that the header is not printed (the header is the part of the response that occurs before the first blank line). Note that the header uses \r\n instead of \n to denote the end of a line. To do this, it may help to put the message body into a single (potentially long) string before printing it. Don’t assume a maximum size for a webpage. Instead, grow your array as needed. (Start with some size. If this fills up, allocate an array that’s twice the size, copy the current contents, and make the next call to recv.)

4. (5 points) Expand the program from the previous part so the list of links is printed after the message body. Each link should be printed in the following form:

    1. text: David Bunde
       URL: http://faculty.knox.edu/dbunde

where the text is the part between the open and closing tags for the link and the URL is the part inside the quotes in the first tag. The link above came from the following html code:

    <a href="http://faculty.knox.edu/dbunde" target="_top">David Bunde</a>

Links should be numbered consecutively starting with 1.

You may assume that there is only a single space between a and href in the link and are not responsible for finding links that violate this. Also assume that “a href” is lower case. There may also be other text in the tags (like the target part above); the link is the part between the first quotes and the text is the part between the first > and the next </a>. Don’t worry about processing HTML comments; it’s fine if you report links that are commented out.

5. (Extra credit: 6 points) Expand the program from the previous part so that after displaying a page, it takes a number from standard input and displays the page pointed to by the link with that number. Links that do not begin with “http” are relative links. These are converted into a URL by using their value to replace the part of the current URL following the last slash.