World Wide Web - WWW

2014 rev 3

The purpose of this lab exercise is to improve your understanding of the basic architecture of the World Wide Web (WWW). After the exercise you should be able to create Web pages, apply HTTP to retrieve header lines and entire Web pages. You should also be able to understand how dynamic Web-pages works. In this lab you will learn how to write in HTML language and do a simple website.

1 Literature

There are a lot of information regarding web pages on the internet. Here are a couple of the most useful links:

- A basic introduction into the WWW:
  `http://en.wikipedia.org/wiki/World_Wide_Web`
- HTML, XHTML, CSS
  - HTML:
    `http://www.w3schools.com/html/default.asp`
    `http://www.htmlcodetutorial.com/`
  - CSS:
    `http://www.w3.org/Style/Examples/011/firstcss.en.html`
    `http://www.w3schools.com/css/`
  - XHTML: `http://www.w3schools.com/xhtml/`
- HTTP
  - HTTP made really easy: `http://www.jmarshall.com/easy/http/`
  - Uniform Resource Locator - URL: `http://en.wikipedia.org/wiki/URL`
  - HTTP session management, including cookies:
    `http://www.it.lth.se/courses/will/monash_session.html`
- PHP
  - PHP tutorial: `http://www.w3schools.com/php/`
  - PHP manual: `http://www.php.net/manual/en/`

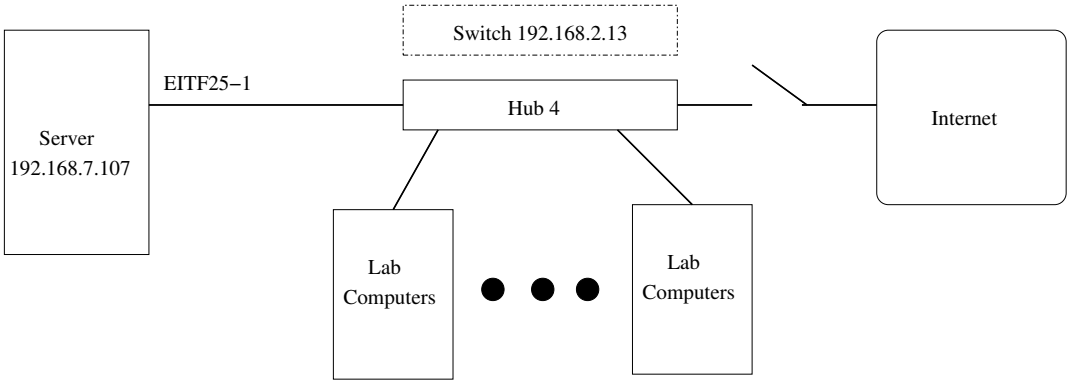

Figure 1: Network configuration during this lab.

2 Theory

We will in this section write a short HTML document. although several of you probably are familiar with HTML code, here is still a small introduction.

2.1 What is HTML?

HTML (HyperText Markup Language) is a language for describing the structure of a Web pages.

Firstly, HTML is not a programming language or protocol in the normal meaning, it is a markup language. As such it consist of a set of markup tags that are used to describe web pages. A browser, e.g. Internet Explorer or Firefox, interprets the HTML code and display a the web page in the desired fashion.

The markup tags in HTML are usually called HTML tags. Their notation is a keyword surrounded by angle brackets like `<html>`. Most often the tags come in pairs like `<b>` and `</b>`, where the first tag in a pair is the start tag and the second tag is the end tag. The start and end tags are also called opening tags and closing tags. An HTML document typically contains HTML tags and plain text in a combination to constitute a web document, or a web page.

For example the following line can be written in a HTML document:

```
<h1> This is my homepage. </h1>
```

Here `<h1>` is the start tag and `</h1>` the end tag. The tags in this case mean that the text "This is my homepage." is a heading of type 1. A subsection is then denoted as `<h2>` and a subsubsection `<h3>`, and so on.

All HTML documents have certain basic standard tags that wrap up the page. These are `<html>`, `<head>`, `<title>` and `<body>` and corresponding end tags. A brief HTML documents with such standard tags may look like below.

```
<html>
<head>
  <title>First Web-page</title>
</head>
<body>
```

```
  <h1> HTML is simple to learn </h1>
  All new students are welcome to LTH
  <br> Here you will learn a lot!
</body>
</html>
```

Below are the tags used in the document above and a few more tags. It does not matter if you use lowercase or uppercase when writing HTML code. `<h1>` will be interpreted in the same way as `<H1>`.

`<HTML>`

This tag tells your browser that the file contains HTML-coded information. Each HTML document begins and ends with this tag.

`<HEAD>`

HTML documents can be fundamentally divided into two parts, a head and a body. The `<HEAD> ... </HEAD>` contains the title, metadata, and information on style sheets and scripts, while the `<BODY> ... </BODY>` contains the markup with the visible content.

`<TITLE>`

The title is the name of the page and displayed in the browser's window bar, it does not appear inside the window. The tag `<title>` have to be in the head part of an HTML document.

`<BODY>`

The tag `<BODY> ... </BODY>` identifies the body of an HTML document. This part contains the information that the browser will show on the screen.

`<H1>`

HTML has six levels of headings, numbered from 1 to 6, where 1 is the one with the largest font. The text enclosed by `<H1> ... </H1>`, is therefore the biggest heading the browser can display. Headings are shown by a browser bold face with a bigger font.

`<BR>`

A browser does not take into account whether you have put carriage returns (CR) in your HTML code. You have to use a tag `<BR>` to break a line and get to the next line. `<BR>` stands for 'BReak'. If we had not used a `<BR>`-tag, the two lines "All new students are welcome to LTH" and "Here you will learn a lot!" would have ended up on the same line, next to each other. You can also mark a paragraph with the tag `<P> ... </P>`.

`<img src="picture.ext">`

With the above command, you can include a picture of your website. Source, 'src', is the path of the source file with the image you want to show, in this case `picture.ext`. Note that the image is not part of the HTML document. Your web browser will download and add the image to the web page separately. Adding `alt="some text"` in the tag gives the browser a text to display if it cannot download the picture file. Then we get

```
<img src="picture.ext" alt="This is my image.">
```

There are meny more options you can set for an image tag, which can be found at e.g. `http://www.w3schools.com/`. If the image is in the same directory as the HTML file, you only write the file-name, otherwise you should also specify a path. Since directory structure in the Web server is not the same as on the computer, it is easiest to store the files that make up a web page in the same directory as the HTML document. You can then use so-called relative addressing, and are not dependent on how the other file structure looks like. Relative addressing is when you enter the path to files starting in the directory where the HTML document is. Lets say that the HTML file is located in the directory

3 Home assignments

Preparation: Read up on all the material in the Literature section and answer all home assignments.

Answer the following questions:

- What is a URL and how is it constructed?
- How is an HTML page sectioned?
- What are the major elements of the "head" part of a Web page?
- What is metadata?
- How can you see metadata in a browser?
- Construct a few HTML Meta-tag lines (assignment 4.3) on paper.
- How do you add a special style to an element when you 'mouse' over it?
- Describe the following HTTP commands: GET, HEAD, POST.
- How can you give a variable to a web page?

4 Lab assignments

4.1 Login

There is a local account, 'telecomuser', with password 'telecomuser', on all machines in the lab-room.

4.2 Web site creation

Start by creating the folder `/var/netwlab/<myWebSite>` (change the folder name '`<myWebSite>`' to something specific for your group) with the command[1]

```
sudo mkdir /var/netwlab/<myWebSite>
```

and make the folder your own with the command

[1] if the folder already exits you can remove it first with the command '`sudo rm -r /var/netwlab/<myWebSite>`'

`sudo chown <username> /var/netwlab/<myWebSite>`
Now you should be able to read and write files in that folder.

Write a simple HTML document and save it as `<me>.html` (change '`<me>`' to for example your first name) in the folder '`<myWebSite>`' in the Web-server root. Make sure that the extension is correct – '`.html`'! Text-editors are available in the menu 'Applications > Accessories' at the top left of the screen.

This HTML document is the foundation of your website. Insert an image in the web-page. Images can be found in the `/var/netwlab/pub/pics` folder. Copy some images to the folder '`<myWebSite>`'.

- Now take a look at how your HTML document looks in a browser. Start Firefox (earth icon in the top bar or via menu 'Applications > Internet') and then open your HTML document entering the URL `http://localhost/myWebSite/<me>.html` in the address bar at the top of the browser.
- Create a small Web site in your folder '`<myWebSite>`' consisting of (at least) two different Web pages (linking to each other), using either XHTML or HTML. The Web site pages should contain at least the following elements: headings, paragraphs, bold text, lists, and both internal (bookmark inside a Web page) and external hyperlinks with anchor-text.
  - What are the URLs of your pages?
  - Check that you can access the pages of another lab group nearby. Ask your neighbours about their addresses.

4.3 Metadata

- What are metadata used for?
- Add the following metadata elements to a Web page you created: keywords, description, and author.

4.4 HTML forms

- Add a form to a Web page you created.
- What kinds of control types exist in forms?
- What is 'hidden' used for?

4.5 Cascading Style Sheets - CSS

Review the CSS tutorial *Starting with HTML + CSS* (from the literature list) at
`http://www.w3.org/Style/Examples/011/firstcss.en.html`

- Take one of your pages (with bold-face text) and add an inline CSS style sheet (in the head section) that make bold-face text use a larger font size. The style sheet can look like this

```
<style type="text/css">
b
 {
```

```
    font-size: 150%;
  }
</style>
```

- Change the style sheet to make the bold-face text red on a blue background.
- Add another piece of text in bold-face, but make it belong to a different class than your previous bold text. (Example `<b class="another">more text to view</b>`) Make text like that (but not your other bold-face text) use a smaller font than normal and be blue on a red background.
- Change from an inline style sheet to an external.

4.6 More CSS (if time is available)

- Design a presentation style for your Web pages using an *external CSS file*. Use a presentation style with three boxes as suggested in Figure 2. Use `<div>` tags with different classes

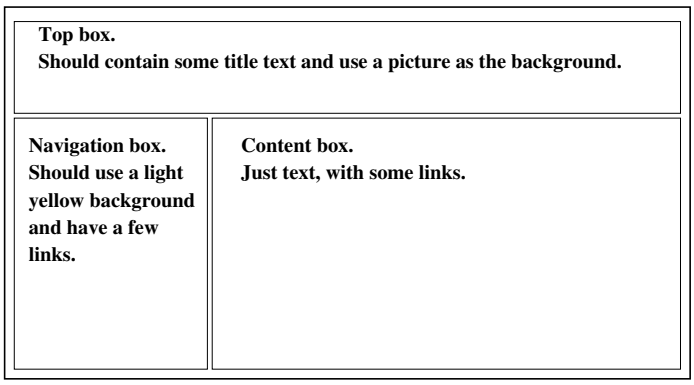

Figure 2: Suggested Web page design.

  to implement the boxes.
- Modify the Web pages to use this presentation style and verify that it works.
- Change the style so that links gets enhanced (bold, color and/or background) when the mouse pointer hovers above it.
- Modify your style so that link enhancement only occurs in the navigation box.

5 Dynamic pages

Most web sites today are built with dynamic pages where the data are stored in a database. This section will give an introduction of how this can be done. There are a variety of techniques by which a Web server can generate content dynamically or "on-the-fly" by feeding the output of programs back to the remote browser. The idea is that the web server uses a program to construct the HTML page, depending on some previous history or input. Notice that the web server always outputs an HTML page. This mechanism is called the Common Gateway Interface (CGI). The programming of CGI scripts can be made in a variety of languages, but today the dominating is PHP in combination with a mySQL database.

5.1 PHP server pages

On the computers in the lab the web server is an Apache server, as in most UNIX or Linux based computers. In PHP special mark-up is used within otherwise normal HTML pages. The special mark-up is interpreted as a program by an Apache plug-in and the generated output is inserted into the normal HTML page at that place. A very simple 'Hi there' page with the wall clock time looks like:

```
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML//EN">
<html>
  <head>
    <title>Hi there</title>
  </head>
  <body bgcolor="#ffffee" text="#000000">
      <h1>Hi there</h1>
        The time is: <b>
        <?php
           date_default_timezone_set('CET');
           echo date('H:i:s');
        ?>
        </b>
  </body>
</html>
```

Notice the part in the tags `<?php ... ?>` which is the PHP program generating the dynamic content.

- Copy the example and try it out. What extension should the program have? How is the program called from the browser?
- When you address the page on your own computer all the client (web browser) calls the web server (Apache). Explain how the program is run and what parts is done where.
- Write a simple PHP script that add two numbers and displays the result.

  For your inspiration here is a simple example of a PHP script that uses an HTML form with one parameter. If it's called without a the parameter 'yname' it just presents a form that allows you to enter a name and then calls itself but with the parameter set to the name. If it's called with the parameter 'yname' set it displays a greeting 'Hi there <yname>'.

```
<html>
  <head><title>Simple form</title></head>
  <body>

<?php
   $yname=$_GET['yname'];
   if ( $yname=='' ) { #empty - no parameters
?>

      <form action="simpleForm.php" method=get>
      Your name:
      <input type="text" name="yname" size="25" value="">
      <input type="submit" value="Hi">
      </form>

<?php
   } else { echo "Hi there $yname\n";}
?>

  </body>
</html>
```

Notice how the if-statement in the PHP code is combined with HTML code. First, the PHP code checks if 'yname' was given. If not an HTML form is given, but if it was given the text "Hi there <yname>" is given.

- How are parameters transferred together with a web page call?

5.2 Basics of a blogg

In this section we will give the ideas behind building a blogg page. Then there should be two parts on the site, one public where the blogg is shown and one administrative where the posts are written. In our case it will mean two pages. On the administrative page a form is used to input the post and writing it to a file. The public page reads the same file and outputs the result on the page. Formally, the only connection between these to pages is that they use the same file for the blogg post, see Figure 3.

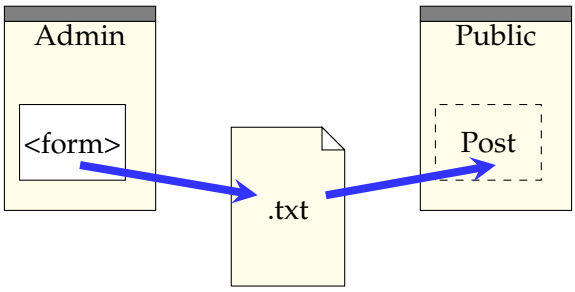

Figure 3: Structure of the blogg. A post is written on the administrative page and saved to a file. Then the same file is read as input to the public page.

Lets start with the public page. Then it is needed to have a file where the post can be saved. In a unix system a file can be created with the command `touch`, and since the directory is outside the user space `sudo` must be used. So in the directory you want the file type something like

```
sudo touch <FileName>.txt
```

Open the file with an editor and write some text in it.

Next, create a new web page with the content you like to have on your blogg. To read a file in PHP the following commands can be used: `fopen`, `fread` and `fclose`, for info see e.g. w3school. The read command requires to input also the number of characters read. For this it can be good to use the command `filesize`.

When the public page works it is time to do the admin page. As a start construct a page with a form where the post can be written. It is a good start to use a form construction corresponding to addition page in the previous section. the form then contains only the text input, and the result displays the text. Together with the display of the text it is also nice to give the text input form again so the same page can be used directly. This means that after the call from the form the text string with the post can be stored in a variable. Then the idea is to write this string to the file, which can be done with `fwrite`.

However, there is now a problem with accessing the file. Since the web server in this case act as any user the access rights of the file must be changed to allow writing. This can be done by

```
sudo chmod a+w <FileName>.txt
```

That should be it. There is a security issue in having a file with write permission in the web server area. To circumvent this the data should be stored at a seperate directory, only available from within the server.

When your one-post blogg works it is time to start thinking of how to extend it. The following questions is not required to be implemented. It should be viewed as discussion questions.

- In the current implementation only the latest post is saved. Lets say you want to save all posts and print the five most resent. How can you achieve this?
- How can you let people comment on public page posts? Are there any security issues with this?
- How can you add pictures to your blogg?

With these requirements the system of files soon are getting hard to overvew. In the real world it is common to use a database to handle the posts and comments. Also all other parts of the web page like header, menus, texts, colors and images can be stored in a database. There are several Content Management Systems (CMS) that can be used. For bloggs it is common to use WorldPress, which is a CMS originally made for blogging. It can also be used to manage non-blog pages. Other common CMSs are Joomla or Drupal and for Wikis the most common CMS is MediaWiki.

6 Clean up

After showing your pages to the lab assistant, clean up by removing the folder '`myWebSite`' and all it's contents.