

System Administration for Beginners

Week 11 Notes

April 17, 2007

1 Lecture on Special Topics #1: HTML

1.1 Where We Stand

We have now finished the majority of the course with respects to and introduction to Beginning UNIX System Administration. In the past ten weeks, we have essentially started from scratch and worked our way up from simple commands on the command prompt to the design and implementation of a reasonably secure Internet server with the capabilities of serving multiple users in an untrusted environment.

1.2 Where We are Heading

System administration is not like your high school Algebra I course. A proficient system administrator is constantly learning about different technologies and keeping up to date with the latest of almost everything and anything in order to stay sharp or maintain marketability. Wherever you decide to go in system administration is up to you. We will have a lecture near the end of the course about the different opportunities and paths in doing sysadmin work.

Thus, it only makes sense to start covering different topics and concepts that you will inevitably hit upon, whether or not you decide to explore further into System Administration. Earlier in the course I handed out a survey asking what kind of topics you may want to see covered. One of the top entries noted was learning how to set up a website. The lecture this week will be covering how to do so in greater detail.

2 Introduction to Websites

We have already seen how to implement the back-end portion of websites. Remember the LAMP model from week 8: GNU/Linux, Apache, MySQL, and PHP. Many websites out there implement this model or some derivative of this model in order to get a website up and running. All of the components can be further generalized to the four essential components: operating system, HTTP daemon, database daemon, and a scripting language. This model provides the

building blocks from which you can use to have websites serving whatever purpose is needed, whether an e-commerce site with a shopping cart or a search engine serving as a library card catalog.

Websites do not spawn and create themselves. Someone needs to create the website and provide methods of interaction between the user and the server through an interface, specifically, the website that the user is visiting. At the moment, we will look at this process as three systems and objects working together.

2.1 A Dynamic Process

2.1.1 The User

We will take the approach from an outsider's point of view. A visitor, or a user, armed with a computer and an Internet connection, will use some sort of software to visit a website. In this example, the visitor is looking to make some sort of purchase on the website at `example.com`. Specifically, he or she will be using an Internet browser, like Mozilla Firefox, to "surf the web". He will type in the address bar the URL of the website (`example.com`) and reach his destination (does anyone remember what role DNS plays here?).

2.1.2 The Server

The server receives the user's connection through the HTTP daemon on a standard port (which one?) and a link is now made between the client (the visitor) and the server. This is the realm in which we have been working for the past ten weeks. Note that the client does not necessarily know anything about the server specifications. There should not have to be any tweaking of settings or special programs (we will discuss external applications, like Adobe Flash later) to specifically access the webpage hosted on your server.

2.1.3 The Website

This is what ties the user and the server together. The website is an interface in which gives information (hopefully) to the user. At this point, the user may interact with the website through data input or entry. At our `example.com` website, the user wants to purchase some goods that Foobar Company is selling. He inputs the information necessary, which the website then organizes through the interaction between itself and the server. There may be an internal website, for example, that Foobar Company can access to see that someone has purchased something and that they now need to take action to ship the ordered product and bill the client appropriately.

2.2 Blackbox Implementation

For the most part, a website designer is not necessarily concerned with the backend system of his or her server. The LAMP model, or whichever model

the system administrator may want to use, should be implemented in a fashion such that the client, a website designer, does not need to worry about changing settings, installing packages, or fussing around with much to get the website to work.

All the designer is concerned about is the ability to get the pages that were designed into the appropriate location such that it can be viewed, if not instantaneously. The designer may be proficient with graphics designing and include a variety of graphics and logos to make the page look interesting and inviting. In all cases, if not most, the designer knows some form of HTML in order to present the website in a fashionable manner.

3 HTML

3.1 What is it?

HTML, or HyperText Markup Language, is simply a special language used by Web browsers (e.g., Mozilla Firefox, Netscape Navigator) to present text and graphics. If you will note, most browsers are able to read plaintext files in a simple manner, but behavior for viewing these files would be governed by the web browser. That is, you may not get a predictable reaction to such a file from all web browsers. Using HTML provides a standard in which most browsers respect.

3.2 History

Tim Berners-Lee, one of the founding contributors to developing the World Wide Web, created the original HTML (and its associated protocols) as a way of facilitating communication of research between himself and his colleagues.

The original design of HTML was simple and made up of twenty-two elements and rather flexible syntax rules to help its adoption by newcomers. As the years go on, HTML specifications were created and maintained by the W3C, or World Wide Web Consortium. They publish specifications which depict syntax rules, elements, and in general, the proper way of going about things when designing a webpage (or browser).

3.3 Versions

HTML is like a living language, undergoing changes to better suit the needs of those who use it (designers and users of the WWW). The first working standard was first published in June 1993, with a revision (HTML 2.0) two years later in November 1995. HTML 3.2 and 4.0 were published at the beginning and end of 1997, respectively, with the current HTML 4.01 at the end of 1999.

It's not necessary to read up on the details of older versions, and probably the best if you use elements and syntax rules from the most current version (much like how no one would publish a modern novel in Old English, if they wanted to target the largest audience possible).

4 Getting Started

4.1 Tags

When working with HTML, anything without markup tags is text that will be presented on the webpage. If you created an entire webpage without using any sort of tags, then it would look rather plain (unless that was your intent). Tags are delimited by the < and > characters. Whatever is inserted between those characters will be recognized as a tag (even if it doesn't exist) and not displayed on the page.

Instead, what it does is modify the text following the tag. For example, let's create a title of a page:

```
<title>My Homepage!</title>
```

The `title` between the < and > characters signify to the browser that whatever follows it should be interpreted as the title of the webpage (usually displayed on the topmost bar of the browser you are running). Note that the tag is also "closed" by repeating the opening tag with a /, or a forward-slash, after the < character. This is known as the closing tag, which should follow every opening tag (with a few exceptions).

4.2 Structure

A webpage generally follows a basic structure, the start of the HTML, the `head` and `body`. Let's take a very basic webpage and analyze it:

```
<html>
<head>
<title>My Homepage!</title>
</head>
<body>
<h1>Welcome! This is the Largest Heading</h1>

<p>The text contained here is located between two
paragraph tags.</p>

<p>Statements can also be <strong>bolded</strong> or
<em>emphasized</em>.</p>

<h2>More Information. The Heading isn't as Large.</h2>
</body>
</html>
```

Here is a list of the basic tags used in the example webpage, followed by its description. Some extra tags are noted in this list as well.

- `<html></html>` : Specifies the start and end of the document. Anything after the closing tag (`</html>`) is ignored.
- `<head></head>` : The first major part of the document. Contains the title, any `meta` tags, JavaScript, stylesheets, etc. that are needed for the formatting of the document.
- `<body></body>` : The second major part of the document. Most, if not all, of the text you'd like to be seen on the webpage goes here.
- `<h1></h1>` : Similar to the built in style for headings in a Word Processor, `h1` specifies the most important (and usually largest) heading in the document.
- `<p></p>` : Specifies a new paragraph.
- `` : Text enclosed will be **bolded**.
- `` : Text enclosed will be *emphasized*.
- `<h2></h2>` : A smaller heading, compared to `h1`. There are 6 levels of headings.
- `<hr />` : Adds a horizontal rule to your page.
- `
` : Forces a line break.

4.3 Adding Interesting Things

4.3.1 Images

A webpage would look rather boring if there were no images or graphics for display. To display an image, we'll first need to have the actual file in the same folder or directory as the HTML document. For example, if I had a file called `mugshot.jpg`, I would use the `img` tag to display it:

```

```

A helpful hint is to organize your website with directories. In this case, let's say I put my `mugshot` file in a directory named `images`. To display the file, I can now reference the picture by referencing the path:

```

```

4.3.2 Links

The Internet wouldn't be known as the WWW if every document was stand-alone. That is, there were no other pages that you could visit from the current page you were accessing. To create a link on your webpage, we will use the `<a>` tags. When referencing a page hosted in the same directory, we will use the tag in the same manner as when we were inserting an image:

Click `here` to learn about me.

Likewise, say we enjoyed learning about Debian GNU/Linux so much that we would like to link to it's homepage. The webpage for Debian can be accessed at `debian.org`, which we will need to link to using its full path:

I love `Debian` GNU/Linux!.

5 Other Things to Note

5.1 Structure-wise

Although mainly semantics, I feel it is a good idea to keep good habits with respects to coding style and formats. Take these things into consideration and you will eventually develop your own style that you're comfortable working with. It is especially important to maintain clean code if you are not the only one working on the website.

- Tags should always be in lower-case.
- Tabbing helps keep the structure neat and understandable by others.
- Try to design the layout on paper first.
- Minimize the use of external applications unless necessary (e.g., Flash, JavaScript)
- Learn from copying. You can view the HTML source of any webpage you visit by going to "View" → "Source" on most modern browsers.

5.2 Editors

There are many editors out there that you can purchase, but most of the time it's better not to be reliant on such editors. Most people stick to Notepad (on Windows), but some editors (like `vim` or `emacs`) have syntax highlighting that will make the job a lot easier.

5.3 Further Resources

There is a whole lot more to web design and HTML than what I could cover in the lecture today. At some places, an introduction to web design is an entire semester's worth of coursework. At this point, you can understand a bit of the language used to create and design webpages, as well as design a very basic and perfunctory webpage of your own (see, for example, the course homepage).

Realize that at times, simplicity works best. Many EECS course webpages aren't filled with anything more than the information that is needed, eliminating images and other bandwidth-hogging resources (which, if you will remember, is not an unlimited resource).

For more information and resources on HTML and web design, check out some of the pages below:

- <http://w3.org/MarkUp/Guide/> - Intro to HTML
- <http://w3.org/MarkUp/Guide/Advanced.html> - Intro Pt. 2
- <http://w3.org/TR/REC-html40/> - HTML 4.01 Specification
- <http://ncdesign.org/html/index.htm> - HTML Design Guide
- <http://alistapart.com> - Style Guides and Articles