CSc 453: Programming Assignment 1 (html2txt): Part 1

CSc 453: Programming Assignment 1 (html2txt): Part 1

Start Date: Tue Sept 1, 2015

Due Date: 11:59 PM, Tue Sept 8, 2015

---

1. General

This assignment involves writing a simple HTML-to-TXT translator. The primary goal of this assignment is to get you acquainted with the compiler front-end tools ***lex*** and ***yacc***, which we'll be using for the rest of the project. A secondary goal is to point out that compiler ideas and tools are applicable for non-compiler problems as well.

In this assignment (Assignment 1, part 1), you are to use the **lex** or **flex** scanner-generator tools to write a program that translates HTML to text. Since the main focus of this assignment is to get you started with these tools, we'll keep things simple and not try to handle all of the subtleties of HTML the way they should really be handled (this means that if you compare your output with the results of a commercial HTML-to-text translator, there may very well be some differences).

Documentation on **lex/flex** is available here.

2.1. Functionality

Your tool should have the following functionality. It should read its input from **stdin**, discard all HTML tags (including comments: see below), recognize and handle a small set of "special entities", and write the remaining text to **stdout**.

2.2. HTML Tags

The HTML standard defines a wide variety of tags. Since the goal of this assignment is to learn to use compiler front-end tools, we will approximate this with a much simpler definition:

> A *tag* is a sequence of characters of the form <*S*>, where *S* is a sequence of printable characters not beginning with a whitespace character and not containing any ">" characters.

Printable characters are specified via the C library function **isprint()**; *whitespace characters* are specified via the C library function **isspace()**. They correspond to the **flex** character class expressions [**:print:**] and [**:blank:**] respectively (see **man flex**).

According to this definition, each of the following is a tag:

```
<b>
<br>
<a href="www.cs.arizona.edu/classes/cs453/fall08/index.html">
</b>
<bgh#i        u&)by 168 jh>

<!@#$%~%^&*()__-+=][;';:,.|>
<!– this is an HTML comment, and has the structure of a
tag as described above –>
```

2.3. Special Entities

HTML defines some "special entities" for some characters. Your tool should recognize the following character sequences in the input and handle them specially, as follows:

Input character sequence	Output
&amp;	&
&lt;	<
&gt;	>
&quot;	"

3. Invoking Your Program

Your executable program will be called **myhtml2txt**. It will read input from **stdin** and write its output to **stdout**. Thus, to translate an HTML file **foo.html** to a text file **bar.txt**, invoke your program as

```
myhtml2txt < foo.html > bar.txt
```

4. Getting Started

To help you get started, I have placed the following files in the directory **/home/cs453/fall15/assignments/html2txt-1** on **lectura**:

- **Makefile**: a sample makefile that shows how **flex** might be invoked.
- **myhtml2txt**: an executable that (supposedly) implements the behavior expected of your program. *This is a x86-64/Linux executable, and will run on **lectura** but may not run on other machines (e.g., Macs or Windows machines).*
- **test.html**: a test input. *Please note that this is **one** of possibly many inputs your program may be tested with; it is made available to be helpful, but does not make any pretense of being exhaustive. It is your responsibility to understand the assignment spec, implement your program accordingly, and test it thoroughly.*

5. Turnin

Turn in your files on host **lectura.cs.arizona.edu**. You should turn in all of your source files, as well as a Makefile that supports the following targets:

clean Executing the command ***make clean*** should delete the ***.o** files, as well as the executable **myhtml2txt**, from the current directory.

myhtml2txt Executing the command ***make myhtml2txt*** should create, in the current directory, an executable file **myhtml2txt** that implements your HTML-to-text translator from scratch, by invoking the appropriate tools (**lex**/**flex**) on the input specifications.

To turn in your files, use the command

```
turnin cs453f15-html2txt-part1 files
```

For more information on the **turnin** command, try **man turnin**. **Note:** The **turnin** command copies the files submitted into another directory. Because of this, programs that compile and execute without problems in your directory may not work correctly once they are turned in, because of problems with relative path names in include files and make files. Such problems are considered to be sloppiness inappropriate in an upper division course, and are liable to be penalized heavily.

The output of your program will be compared with our output using **diff** utility (see **diff**(1)), so it is recommended that you follow the specification, and instructions for turnin, closely.