Tecnológico de Costa Rica

Curso: IC-3101 Arquitectura de Computadores

Profesor: Erick Hernández Bonilla

Alumnos:

Adrián Garro Sánchez 2014088081

Liza Chaves Carranza 2013016573

Marisol González Coto 2014160604

Primer Proyecto: Validador de Documentos

Fecha de Entrega: 17 de abril

I Semestre 2016

Tabla de Contenidos

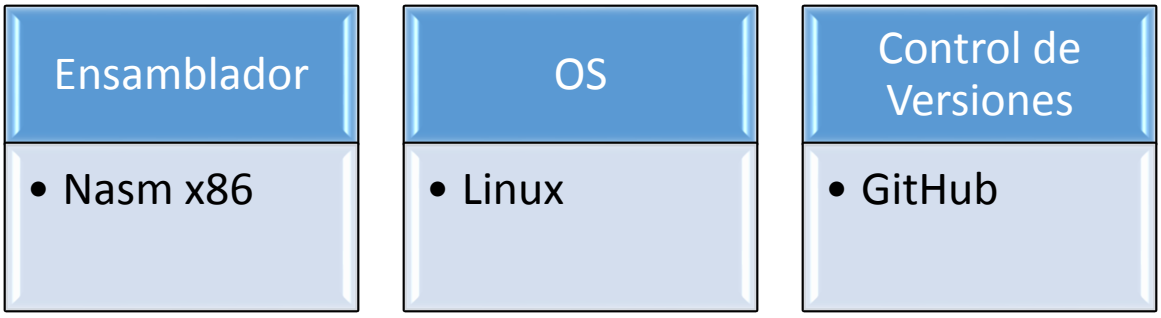
Introducción

El desarrollo web ha tenido un apogeo constante en los últimos años, haciendo que nuevas tecnologías surjan para crear nuevas ideas y creativas páginas en el internet. Estas se basan en varios lenguajes, frameworks o formas para diseñar el contenido de las páginas, que utilizan lenguajes como CSS (Cascading Style Sheets), HTML (HyperText Markup Language) y XML (eXtensible Markup Language) siendo HTML el armazón, CSS el que le da los estilos, tipos de letras, colores, formas, y más y el XML el que transporta los datos de la página web.

La primera tarea asignada para el curso de Arquitectura de Computadores es crear un validador de documentos HTML y XML, que trabajan con *tags*. Estos objetos se abren y se cierran y tienen una estructura específica. Su posición, su apertura y cierre, su indentación y nombres es lo que se pretende validar con el programa a desarrollar.

Arquitectura

El programa se va a desarrollar bajo la siguiente estructura:



Algoritmos de Verificación de Consistencia

Apertura y Cerradura de Tags

El algoritmo para verificar la consistencia de los tags, está separado en varias secciones:

1. Verificar la apertura y cerradura de los tags:
    a. El primer paso consiste en verificar el caracter “<” que es el comienzo de un tag.
    b. Cuando se encuentra este, se sigue el procesxo para buscar el cierre, el carácter “>”.
    c. Si se encuentra “<”, y encuentra “>”, proseguirá con los demás tags para encontrar el tag que cierra. Si no encuentra “>” para cerrar el tag, pero se encuentra otro “<” mostrará un error con la fila y columna correspondiente.
    d. Si no se encuentra “</”, mostrará un error en pantalla, con su fila y columna correspondiente. Si encuentra “</” pero no encuentra “>”, mostrará un error con la fila y la columna correspondientes.

Comillas

El algoritmo es similar al anterior, se revisa dónde comienza la comilla hasta la siguiente, solo que en esta fase, en las comillas existe una restricción de los caracteres que pueden ir dentro de ellas. Según la tabla ASCII, los caracteres que pueden ir dentro de las comillas van desde el 45, que corresponde al caracter “-” hasta el 122, que sería la zeta minúscula. Este proceso aplica tanto para comillas simples como para comillas dobles.

Tags Anidados

El algoritmo para la revisión de tags anidados se basa en una extensión del algoritmo para la verificación de apertura y cerradura de tags. Esta extensión consiste en copiar el contenido de un tag previamente verificado, seguidamente se busca en forma exhaustiva otro tag que posea el mismo contenido que el copiado originalmente, una vez que se comprueba esto se procede a verificar que este tag final posea el símbolo '/' después su inicio (<). Si no se llega a encontrar este tag final, se informará en la consola este error con la línea y columna donde este se generó.

Indentación de Líneas

El algoritmo de identación tanto de los archivos XML como de los archivos HTML se realiza de la siguiente manera:

En primer lugar, se recorre nuevamente el buffer en el que se almacenó el archivo original. Este buffer será entonces recorrido por un proceso de formato para el archivo.

Dado que el archivo de entrada puede estar identado, se idea un algoritmo que prepare el texto para la identación final. Este proceso elimina todos los caracteres de cambio de línea y las tabulaciones que se encuentren en el archivo original. Por lo tanto, el resultado final de este proceso será un archivo sin espacios entre las etiquetas y sin caracteres de cambios de línea.

Con este nuevo buffer procesado, se inicia el algoritmo de identación. Se cuenta con dos registros indices: R11 para el buffer sin identar, y R13 para el buffer identado. También se utiliza R12 como contador de etiquetas, y R14 como contador de dígitos totales en el buffer.

Este algoritmo consiste en recorrer el buffer creado en la etapa de formato. El primer paso es comparar el indice utilizado en el buffer por identar, con el numero total de caracteres leídos. Si estos son iguales, se imprime el buffer con identación. Si no son iguales, se analiza el buffer. La primera comparación es del presente caracter con el caracter de boquilla abierta ('<'). Si son iguales, se guarda en el registro R15 la dirección del proximo caracter para verificar después si esta es una etiqueta de apertura o de cerradura. Seguidamente se compara este proximo caracter, y si es una diagonal ('/'), se disminuye el contador de etiquetas, se agrega un cambio de linea y se llama el procedimiento para agregar los espacios en blanco correspondientes. Este proceso será descrito próximamente. En el caso de que esta etiqueta no esté seguida por una diagonal, se incrementa el contador de etiquetas, se agrega un cambio de línea y se invoca al procedimiento para agregar los espacios correspondientes.

El algoritmo para agregar los espacios en blanco consiste en multiplicar el contador de etiquetas por tres cada vez que se necesite agregar espacios en blanco, y crear un ciclo en el que se agregue al archivo identado un espacio en blanco, hasta que se hayan agregado los espacios correspondientes, indicados por el contador.

Después de agregados los cambios de línea y espacios, se aumentan los índices R11 y R13, y se continúa analizando el buffer hasta que se alcance el tamaño almacenado al principio del algoritmo.

Revisión de Comentarios

La revisión de comentarios se lleva a cabo analizando la estructura de los tags, para que cumplan con la estructura de los comentarios, específicamente esta *<!—comentario -->.* Cuando se encuentra un tag que abre, se revisa si el caracter siguiente es un signo de admiración, si no lo es, sigue buscando caracteres hasta encontrar esto de nuevo, si lo es, verifica si lo que sigue es un guión, si no es un guión, se imprimirá un error en la pantalla, si lo es, verifica el mismo paso, revisando si hay otro guión. Esta primera parte revisa la apertura del comentario. Cuando se encuentra la estructura <--*!* Exitosamente, se procede a buscar un guión para la estructura de cerradura, si no se encuentra, sigue buscando los caracteres, si lo encuentra, verifica que el siguiente también sea un guión, luego busca el caracter de cerradura "<" luego de los guiones y si no lo encuentra da un error, y si lo encuentra prosigue revisando el documento en busca de otros comentarios. Si el comentario no cierra, entonces todo lo siguiente a la apertura del comentario se ignorará y la revisión llegará a su fin.

Manual de Usuario

El programa se compila de la siguiente manera desde la línea de comandos:

$>cd carpeta_donde_esta_el_programa

$>make

Luego se ejecuta de esta manera:

- Para XML:

  $>documents_analyzer < validate_archive.xml

- Para HTML:

  $>documents_analyzer < validate_archive.html

Para cada uno de los casos, se mostrarán los errores que contenga el archivo ingresado, con su respetiva justificación, asímismo la línea y la columna donde se dieron. Si el archivo no posee errores, se mostrará en la pantalla el nuevo archivo con las identaciones necesarias, si aplica.

Bibliografía

NASM. *NASM Manual.* Recuperado el 06 de mayo de 2016, desde http://www.nasm.us/doc/

Wikibooks. *X86 Assembly/Control Flow - Wikibooks, open books for an open world.* Recuperado el 07 de mayo de 2016, desde http://en.wikibooks.org/wiki/X86_Assembly/Control_Flow

Code as Art. *Code as Art: Assembly x86_64 programming for Linux.* Recuperado el 08 de mayo de 2016, desde http://0xax.blogspot.com/p/assembly-x8664-programming-for-linux.html