

Primer Proyecto de Compiladores

Analizador Léxico

Realizado por:

Luis Prado

Joseph Araya

1 Semestre 2013

Contenido

Descripción del problema:	3
Diseño del programa:	3
aumentarFilaColumna():	3
substr():	3
escribirErrores ():	3
Lista de tokens a manejar:	3
T_AbrirEtiqueta:	3
T_CerrarEtiqueta:	3
T_Etiqueta:	3
T_Atributo:	4
T_Texto:	4
Análisis de objetivos	4
Manual de usuario:	5
Conclusiones personales:	5

Descripción del problema:

Esta tarea programada busca la creación de un analizador léxico que logre reconocer XHTML, así se reconocerán las diferentes etiquetas válidas y se identificarán los caracteres no permitidos en las diferentes secciones del documento, además se debe indicar el lugar en donde estos ocurran, se debe considerar guardar los datos importantes que posteriormente se le pasarán al parser en el siguiente proyecto.

Diseño del programa:

Para la implementación del analizador léxico, se utilizó flex debido a que existe más documentación para este que para jflex, además se consideró más fácil de utilizar.

Algunas de las funciones extra que se utilizaron fueron:

aumentarFilaColumna(): Esta función trabaja con `yylen` para llevar el control de la fila y columna, que se utilizan a la hora de reportar errores.

substr(): Esta función se usa para obtener una sub cadena a partir de otra.

escribirErrores (): Esta función se usa para escribir los datos del `stderr` en un archivo de informes de errores, el cual se puede cambiar en el programa, y por defecto se llamará “errores.txt”, en este archivo indicará la fila y columna de cada error.

Además cabe mencionar que se hizo uso de “Start Conditions” para controlar las diferencias entre tokens y texto por ejemplo.

Lista de tokens a manejar:

T_AbrirEtiqueta: Este token corresponde con “<” y se utiliza para saber que después de él tiene que venir un token válido.

T_CerrarEtiqueta: Este token corresponde con “>” y se utiliza para saber que después de él tiene que venir texto.

T_Etiqueta: Corresponde con las etiquetas válidas para `xhtml`

Las etiquetas consideradas son las siguientes:

?xml	A	abbr	acronym	address	applet	area	article	aside	audio
B	base	basefont	bdi	bdo	big	blockquote	body	br	button

canvas	caption	center	cite	code	col	colgroup	command	datalist	dd
del	details	dfn	dialog	dir	div	dl	dt	em	embed
fieldset	figcaption	figure	footer	font	form	frame	frameset	h1	h2
h3	h4	h5	h6	head	header	hgroup	hr	html	i
iframe	img	input	ins	kbd	keygen	label	legend	li	link
map	mark	menu	meta	meter	nav	noframes	noscript	object	ol
optgroup	option	output	p	param	pre	progress	q	rp	rt
ruby	S	samp	script	Section	select	small	source	span	strike
strong	style	sub	summary	sup	table	tbody	td	textarea	tfoot
Th	thead	time	title	Tr	track	tt	u	ul	var
video	wbr								

T_Atributo: Conformado por identificador de atributo en minúscula mas “=” mas comillas mas texto mas comillas, se usa para identificar atributos de las etiquetas xhtml.

T_Texto: Texto entre etiquetas con caracteres válidos

T_Espacio: Representa un “ ”.

T_NuevaLinea: Representa un “/n”

Análisis de objetivos

Rubro	Estado alcanzado
Identifica etiquetas xhtml correctamente	Completado
Uso de stderr para manejo de errores	Completado
Indica fila y columna para indicar errores	Completado
Se utilizó MAKEFILE	Completado
Se presentaron pruebas funcionales	Completado

Manual de usuario:

Para la compilación del programa debe estar en una distribución de Linux, preferiblemente Ubuntu, asegúrese de que flex se encuentra instalado, sitúese dentro de la carpeta en donde se encuentre el proyecto mediante terminal, una vez hecho esto digite make para compilar el programa, para ejecutar el programa digite. ./scanner <Archivoxhtml en el terminal, en donde el parámetro Archivoxhtml corresponde con el archivo a analizar, tome en cuenta que la descripción de los errores se almacenaran por defecto en un archivo llamado errores.txt

Conclusiones personales:

Joseph Araya Rojas: El desarrollo de analizador léxico me permitió comprender mejor cómo funcionan los compiladores en la primera etapa de su análisis, además de entender que tan complejos pueden y tienen que ser estos para poder manejar un determinado lenguaje e indicar los errores cuando estos se presentan.

Luis Prado Rodríguez: Al realizar esta primera etapa del proyecto pude comprender mejor como funciona el lenguaje xhtml y como se debería hacer un scanner para este, además de entender cual es la principal funcionalidad de los scanners. Me pareció un proyecto interesante y espero que las siguientes partes lo sean más.