



Tarea #5

Lex&Yacc

IC-5701 **Compiladores e Intérpretes**

Instituto Tecnológico de Costa Rica, Sede Central Cartago

Escuela de Computación, Ingeniería en Computación

I Semestre 2015

Prof. Esteban Arias Méndez

Para el desarrollo de Compiladores e Intérpretes el hacer uso de herramientas que ayuden en una mejor estructuración del software y faciliten su mejoramiento y mantenimiento, las herramientas tales como Lex & Yacc son aliados fundamentales en esta labor.

INTRODUCCIÓN

Propósitos:

- Esta tarea tiene como objetivo principal el familiarizarse con las herramientas Lex&Yacc (o las versiones más recientes y para el lenguaje de programación que usted utilice), estas herramientas le serán de gran utilidad en la implementación del Proyecto del Curso y para sus 3 etapas y entregables.
- Primeramente, haremos un ejercicio que consiste en la lectura de un manual sobre un pequeño intérprete desarrollado en Lex&Yacc con el lenguaje C basado en una calculadora sencilla. Luego de leer el manual, se tomarán los archivos fuentes y se deberán procesar para generar los programas de salida requeridos. La principal utilidad de esto será el tener disponible archivos de ejemplo funcionales escritos para Lex&Yacc.
- Con las descripciones básicas de los archivos de ejemplo y su comprensión, procederemos a escribir los archivos de entrada necesarios para la implementación de un pequeño lenguaje denominado MICRO para el cual se brinda la gramática en formato BNF. Acá se espera que utilice las mismas herramientas que usará para su proyecto para poder usar este ejercicio como práctica y prueba para el desarrollo del proyecto.

DESARROLLO

Esta es una tarea para trabajarse en parejas de 2 máximo. Los siguientes ejercicios Parte I y Parte II deben desarrollarse para trabajar con la herramientas Lex&Yacc (o sus versiones equivalentes) sobre el sistema operativo LINUX. Para la primera parte en el lenguaje C, en el cual está desarrollado el ejemplo y para la segunda parte usando el lenguaje de programación que usará en su proyecto

Parte I

En el link <http://epaperpress.com/lexandyacc/> pueden encontrar un tutorial de donde vamos a realizar esta primera parte.

Primeramente se tomará el contenido de dicho sitio web como una lectura, para lo cual deberán preparar un esquema/resumen de la teoría acá expuesta como marco teórico de la documentación a entregar. Esto servirá como guía para el trabajo a desarrollar con las herramientas.

La tarea consiste luego en conocer el uso de las herramientas Lex & Yacc como generadores de Scanner y Parsers respectivamente. Comprender su funcionamiento interno y aprender a usarlas, lo cual será de utilidad para el proyecto.

Para esto, se debe tomar el código fuente sobre una calculadora que se analiza en el link dado, para procesarlo con Lex & Yacc.. y compilar el programa generado en C y realizar pruebas con diversos ejemplos.

En la carpeta del material digital del curso también se encuentra la información del link dado. Poner especial atención a los ejemplos brindados y la forma de uso de los mismos. En el link se encuentran dos prácticas cortas que pueden emplear como base y guía !

Deben enviar todos los archivos usados y muestras de la ejecución y salidas obtenidas.. así como cualquier prueba realizada fuese funcional o no documentadas en el PDF a ser enviado.

Parte II

En esta parte deberán desarrollar la primera parte de un pequeño compilador para el lenguaje MICRO. Este consiste en obtener un scanner y un parser. La programación debe realizarse usando las herramientas Lex&Yacc en las versiones para el lenguaje de programación que usará en su proyecto del curso. En caso contrario cada una de las partes programadas sería desarrollada manualmente por cada uno. (i.e. escribir manualmente el scanner y el parser).

MICRO es un lenguaje con las siguientes características:

- el único tipo de datos es de tipo entero o integer
- los identificadores son declarados implícitamente y tienen a lo más 32 caracteres
- los únicos literales son hileras de dígitos (0-9)
- los comentarios empiezan con -- y terminan con el fin de línea
- solo hay dos tipos de instrucciones: asignaciones y operaciones de Entrada/Salida
- begin, end, read, write son palabras reservadas
- todas las instrucciones terminan con ';' (punto y coma)
- el programa está delimitado por las palabras reservadas begin y end

La gramática del lenguaje micro es la siguiente: Extended CFG Defining Micro

```
<program>      ->  begin <statement list> end
<statement list> ->  <statement> { <statement> }
<statement>    ->  ID := <expression> ;
<statement>    ->  read ( <id list> ) ;
<statement>    ->  write ( <expr list> ) ;
<id list>      ->  ID { , ID }
<expr list>    ->  <expression> { , <expression> }
<expression>   ->  <primary> { <add op> <primary> }
<primary>      ->  ( <expression> )
<primary>      ->  ID
<primary>      ->  INTLITERAL
<add op>       ->  +
<add op>       ->  -
<system goal>  ->  <program> SCANEOF
```

Para esta parte deberán crear los archivos de definiciones en el formato de sus herramientas Lex&Yacc equivalentes (es decir los archivos "definiciones.y" y "gramatica.y" o similares) ejecutar sus herramientas con dichos archivos para generar el scanner y el parser de MICRO y compilarlos en su lenguaje de programación de trabajo. Deberá documentar el proceso realizado, realizar y documentar pruebas y mostrar ejemplos de uso. Así como incluir todos los archivos usados como apéndices en el documento y enviarlos en digital.

DOCUMENTACIÓN

Se debe entregar un documento en PDF como informe final de la actividad realizada.

El mismo debe incluir:

- El marco teórico que debe incluir la definición y descripción de las herramientas Lex&Yacc y el resumen/esquema del material dado para la primera parte.
- Para el marco teórico puede ayudarse con material de internet, de las herramientas propiamente usadas para Lex&Yacc en C para la primera parte y para las herramientas según su lenguaje de programación para la segunda parte. Además en las lecturas del libro de Louden asignadas hay material de referencia y adicionalmente un conjunto de lecturas breves disponibles en la carpeta "Compiler Design ReferencePoint Suite book" en el material digital del curso, en la carpeta del "Compiladores" dentro de "@ Libros". No olvide las referencias bibliográficas en el texto. (10pts)
- Análisis de resultados de la implementación de la calculadora y los cambios o ajustes realizados para su funcionamiento si los hizo.
- Los archivos de configuración escritos para MICRO y explicados en el documento.
- Pruebas de ejecución exitosas y fallidas para ambas partes.
- Apéndices con el texto de los archivos más importantes que haya trabajado.
- Indicar las herramientas utilizadas, indicar cómo las obtuvo e instaló y usó.
- El documento debe estar completo con su portada con todos los datos completos del curso, etc. e incluyendo un abstract en inglés en la misma portada, conclusiones y observaciones finales y bibliografía usada (10pts).

EVALUACIÓN Y ENTREGA

1. Rúbrica de evaluación.
 - 25% Documentación
 - 25% Primera Parte
 - 50% Segunda Parte

Cada parte se calificará con los siguientes criterios:

- a. 50pts - Funcionamiento correcto del programa según lo solicitado.
 - b. 20pts - El código internamente implementa la solución como se solicitó y no de otra forma
 - c. 10pts - El código se encuentra documentado
 - d. 10pts - Mensajes informativos al usuario son apropiados y claros
 - e. 10pts - La implementación es eficiente
2. Cada parte debe resolverse, implementándolo de la mejor manera.
 3. Se revisará el código de los programas para revisar que se cumpla con lo solicitado de forma interna y no solo el comportamiento del programa sea como el solicitado.
 4. De forma global, se evaluará la presentación del trabajo según los parámetros solicitados, la entrega a tiempo del trabajo y la documentación debida.
 - a. 2pts - El subject del correo a ser enviado debe ser:
[CO] – Tarea # 5 - SusNombresCompleto
 - b. 2pts - El correo debe contener de forma separada 3 archivos. Los archivos trabajados para cada parte en un comprimido .zip, 1 por cada parte por separado; y un tercer archivo, la documentación en PDF con el nombre de archivo igual al subject del correo enviado. NO envíe archivos ejecutables o binarios.
 - c. 5pts - La documentación debe incluir una portada con los datos completos: Universidad, carrera, sede, curso y código, profesor, periodo, fecha de entrega, número de tarea, título de la tarea, nombres completos con número de carnet de la pareja de trabajo y un abstract en inglés en la misma portada.

- d. 10pts - La introducción del documento es una descripción breve del trabajo realizado y herramientas usadas.
- e. 10pts - Marco teórico según se ha explicado.
- f. 10pts - Como desarrollo deberá describir la forma en que fue solucionado cada ejercicio, indicar los ejemplos de código que ha usado como guía para el desarrollo de los mismos usando las referencias bibliográficas correspondientes. Explicar el uso y funcionamiento, y mostrar capturas de pantalla que muestren ejemplos de la ejecución de cada ejercicio.
- g. 10pts - Análisis de resultados.
- h. 5pts - Para cada parte documente su estado: completado con éxito, completado de forma incompleta, no funcional, no implementado, etc. Y las razones o justificaciones en cada caso.
- i. 5pts - Como una sección de conclusiones y/o observaciones indique en 1 párrafo, máximo dos, sobre cada parte, problemas presentados y si tiene alguna opinión del mismo.
- j. 5 pts - En una sección de Apéndices incluya el código fuente de cada ejercicio utilizando un tipo de letra como courier # 8, para que el código sea legible y bien indentado en el documento.
- k. 10pts - Incluya comentarios en el código y los archivos de configuración de las herramientas para presentar de forma correcta el trabajo.
- l. Se debe entregar en digital, según se indicó a más tarde el Lunes 27 de Abril, antes de la media noche. Debe hacerlo de forma simultánea a los correos siguientes y copiarse usted mismo y su compañero de trabajo. Cada día de atraso serán 10pts menos de la nota de la tarea.
 - i. clases.esteban@gmail.com
 - ii. jodytailee@gmail.com
- m. Cualquier consulta puede hacerla al foro, o personalmente al correo del profesor con copia al asistente a los correos anteriores.
- n. 10pts - La tarea será revisada posteriormente con citas de revisión o en clases. Debe tomar en cuenta que se le harán preguntas orales sobre el código durante la revisión, por lo que esto será parte de la evaluación de la tarea, para un máximo de 10pts. Durante la revisión deben estar presentes ambos miembros de la pareja de trabajo, la no presentación a la cita de revisión les restará los 10pts de revisión a los ausentes.