

Compiladores

Proyecto de Curso

Generalidades

El proyecto a desarrollar durante este curso consiste en la implementación de un compilador que permita generar código ensamblador para un procesador MIPS R2000, a partir de módulos escritos para Visual Basic 6.0.

Para correr los programas en MIPS se utilizará el simulador PCSPIM con capacidad de correr en un ambiente Windows.

Se utilizarán también los generadores JFlex y CUP para la implementación del compilador y el lenguaje a utilizar será Java.

El proyecto se desarrollará en dos fases, una en cada curso de la clase de Compiladores, éstas se describen a continuación:

Fase No.1

La primera fase del proyecto consiste en crear el analizador léxico y sintáctico. El proyecto a presentar debe considerar:

- Desarrollar una gramática que permita reconocer programas que corran en Visual Basic con las siguientes características:
 - o Que puedan utilizar funciones y procedimientos con un número ilimitado de parámetros de los tipos básicos, por referencia o por valor.
 - o Debe reconocer los tipos Integer, Boolean, Char y String
 - o Las declaraciones de variables podrán escribirse dentro de cada estructura (bloque), función o procedimiento, o fuera de éstos.
 - o Debe incorporarse en la gramática las funciones para lectura y escritura.
 - o Deben reconocer los bloques If, For (en su forma más sencilla) y Do While...Loop
- Se permite la recursividad.
- Deberá manejarse la precedencia de operadores común, es decir + y – tienen precedencia más baja que * y /. De igual forma, el operador = tiene precedencia más baja que el resto de operadores relacionales. Lo mismo para operadores booleanos
- Debe ser capaz de reconocer y recuperarse de errores léxicos y errores sintácticos. Se deberá imprimir la línea y columna donde fueron encontrados cada uno de éstos y una descripción útil del mismo.
- El analizador sintáctico deberá crear un AST que será utilizado en las siguientes fases del proceso de compilación.

- Se entregarán todos los programas fuente; así como los archivos .class o .jar. El programa final deberá consistir en un solo programa que reciba como parámetros un archivo .vb. Deberán entregarse además, 3 archivos fuente correctos y 3 con error.
- El programa deberá tener un main, de lo contrario imprimirá error.

Fase No.2

La fase No.2 consiste primordialmente en la generación del archivo .asm que correrá en el PCSPIM. Consideraciones:

- Deberán implementarse las funciones para lectura y escritura de variables, para todos los tipos básicos de la gramática
- Los parámetros a funciones podrán ser pasados por referencia o por valor.
- Manejo correcto del ámbito tal como lo maneja Visual Basic 6.0.
- Ningún identificador deberá ser declarado dos veces en el mismo ámbito.
- Ningún identificador podrá ser usado sin ser declarado previamente en el ámbito correspondiente.
- Chequeo de tipos. No se permiten asignaciones ni comparaciones entre variables o constantes de diferentes tipos. También deberán chequearse que los parámetros enviados a las funciones sean el número y tipo correcto.
- Retorno de valores solamente en funciones.
- Generación de código intermedio.
- Generación de código MIPS
- Se entregarán todos los programas fuente; así como los archivos .class o .jar. El programa final deberá consistir en un solo programa que reciba como parámetros un archivo .vb y generará un archivo .asm a ser ejecutado en el simulador SPIM. Deberán entregarse además, 3 archivos fuente correctos y 3 con error; así como un reporte especificando las técnicas de programación utilizadas, los problemas encontrados y la solución encontrada a cada uno de ellos.

Convenciones Lexicográficas

- Comentario es toda secuencia de caracteres que se encuentra después de una comilla (') en una línea.
- Las sentencias se separan por carácter de nueva línea
- Las cadenas de caracteres podrán extenderse a través de varias líneas añadiendo un espacio seguido de &_ al final de la línea. Por ejemplo:

Cadena = "Introduzca el nombre," &_
" luego introduzca su fecha de nacimiento"

Puntos Extras

- Implementación del tipo variant
- Creación de ejecutable para un procesador Intel moderno.