

Definición general

Esta última etapa del proyecto conocida formalmente como Generador de código es, si se quiere la más significativa del proyecto, pues reúne las etapas anteriores y termina el compilador

Para esta etapa del proyecto se debe entregar un programa que reciba un código fuente escrito en Nuestro Pascal y realice el análisis léxico, sintáctico, semántico y genere código ensamblador. Para esto se debe utilizar la definición del lenguaje aceptado realizado en la primera etapa del proyecto junto con la gramática diseñada en la segunda parte del proyecto

Se espera que la traducción del código se lleve a cabo mediante la Traducción dirigida por Sintaxis, por lo que se recomienda seguir los ejemplos vistos en clases con lo relacionado a las acciones semánticas y los registros semánticos correspondientes.

Al finalizar el análisis el programa deberá desplegarle al usuario el resultado del Análisis léxico, sintáctico y semántico que se efectuó. Se espera que despliegue:

1. Listado de errores léxicos encontrados: El programa debe desplegar una lista de todos los errores léxicos que se encontraron en el código fuente. Debe desplegar la línea en la que se encontró el error.

Es importante que el programa deba poder recuperarse del error y no desplegar los errores en cascada ni terminar de hacer el scaneo al encontrar el primer error.

2. Listado de errores sintácticos encontrados: El programa debe desplegar una lista de todos los errores sintácticos que se encontraron en el código fuente. Debe desplegar la línea en la que se encontró el error. Además, el mensaje de error debe ser lo más específico posible, con el fin de que el programador pueda llegar al error y corregirlo de forma eficiente. Es importante que el programa deba recuperarse del error y evitar no desplegar errores en cascada ni terminar de hacer el parseo al encontrar el primer error.

3. Listado de errores semánticos encontrados: El programa debe desplegar una lista de todos los errores semánticos que se encontraron en el código fuente. Debe desplegar la línea en la que se encontró el error. Además el mensaje de error debe ser lo más específico posible, con el fin de que el programador pueda llegar al error y corregirlo de forma eficiente. Recuerde que en el momento en que ocurren errores no es necesario que continúe traduciendo pero si debe continuar analizando el código tanto en su léxico, como en la sintaxis y en la semántica.

4. Contenido de Tabla de Símbolos: Desplegar el contenido de la tabla de símbolos una vez concluido el análisis. Debe especificar el nombre, tipo y ámbito de variables globales, funciones y variables locales y parámetros.

5. Archivo con Código Ensamblador: Si no ocurrieron errores durante la compilación el programa debe generar un archivo con el mismo nombre del que se compile (cambie la extensión!!). Este archivo contendrá la traducción de ciertas partes del programa solo que en lenguaje ensamblador. Puede generar Ensamblador para Windows o para Linux.

Descripción Detallada

Se les sugiere tomar en cuenta los siguientes aspectos para asegurar la completitud del programa.

ANALISIS SEMANTICO

- Deben realizar análisis Semántico de los siguientes casos.
 - Variables no definidas
 - Variables doblemente definidas
 - Funciones no definidas
 - Corroborar tipos en asignaciones y tipos de operadores con los operandos
 - Corroborar cantidad y tipo de parámetros.
 - Break y continue dentro de los bloques permitidos.

TRADUCCION DE CODIGO

- Deben realizar la traducción de código de los siguientes casos.
 - Declaración de variables globales
 - Declaración de funciones (solo la definición de donde a dónde va la función)
 - Expresiones binarias aritméticas
 - If-else
- Debe implementar la Tabla de Símbolos (no se complique en la estructura de datos a utilizar) y la Pila semántica

Documentación

Se espera que sea un documento con el resultado de la programación, qué sirve, qué no sirve.

Deben especificar, además, como compilar su proyecto. La documentación no se evaluará pero si debe presentarla en la revisión.

Aspectos Administrativos

- Los grupos deben permanecer iguales a las otras etapas del proyecto.
- El trabajo se debe de entregar el día 21 de Junio de 2016. Deben enviarlo por correo electrónico antes de las 12 de la noche, al correo emarin@ic-itcr.ac.cr y erikams79@gmail.com
- Deben entregar el código fuente junto con el ejecutable.
- Las revisiones serán el día siguiente.
- Se debe de entregar la documentación IMPRESA para ser calificado, así como una copia del archivo en el correo. La documentación impresa se debe entregar el día de las revisiones.
- Recuerde que oficialmente no se recibirán trabajos con entrega tardía.