

Proyecto 3: Análisis Semántico y Generación Básica de Código

I. DESCRIPCIÓN

Este proyecto se construye sobre los proyectos anteriores del semestre, agregando capacidades básicas de análisis semántico y de generación de código. Toda la programación debe realizarse en C sobre Linux, usando las herramientas `flex` y `bison`. No se pueden cambiar las especificaciones de este documento.

Su programa recibe como entrada un programa fuente, aparentemente escrito en lenguaje C, y produce como salida un reporte en consola de **todos** los errores sintácticos detectados junto con ciertos errores semánticos. Si el programa no tiene errores se generará el lenguaje ensamblador correspondiente, el cual será ensamblado y ejecutado.

II. EJECUCIÓN

El programa se invocará desde la línea de comando de una consola. Recibirá como argumento el nombre del archivo fuente a ser procesado, junto con cualquier opción necesaria para ejecutar el programa al estilo tradicional de UNIX/Linux. Si el usuario no da los argumentos, o si estos contienen errores (en cantidad o forma) se desplegará una ayuda explicando las opciones disponibles.

III. PREPROCESO Y SINTAXIS

Todas la funcionalidad solicitada en los proyectos previos debe estar presente en este proyecto (*e.g.*, preproceso, análisis léxico, análisis sintáctico, etc.).

Trabajo extra opcional 1: Manejar macros con parámetros

IV. ERRORES SEMÁNTICOS

Se debe manejar una Tabla de Símbolos para reportar los siguientes errores:

- Variables no declaradas
- Variables declaradas múltiples veces dentro del mismo contexto

Trabajo extra opcional 2: Reportar inconsistencias entre prototipos, definiciones y usos de funciones.

Trabajo extra opcional 3: Reportar usos incompatibles de tipos de datos.

V. GENERACIÓN DE CÓDIGO

Se debe generar código ensamblador x86 para las siguientes construcciones:

1. Expresiones aritméticas (enteros)
2. `if then else`
3. `while`
4. `for`
5. Operador ?
6. Definición de funciones
7. Invocación de funciones
8. Expresiones aritméticas (flotantes)

Una semana antes de la entrega del proyecto se establecerá hasta cuál de los puntos de la lista anterior se deben entregar, siendo el resto de la lista trabajo extra opcional.

Cada grupo debe tener bastantes ejemplos de prueba para cada punto de los obligatorios.

VI. EJECUCIÓN DE CÓDIGO GENERADO

El ensamblador x86 generado de acuerdo a la especificación previa quedará en un archivo en el mismo directorio que el fuente original del programa escrito en C. Internamente (ver uso de `system()`), su programa mandará a ensamblar el código respectivo y lo pondrá a ejecutar. Todo sobre Linux.

VII. REQUISITOS INDISPENSABLES

La ausencia de uno solo de los siguientes requisitos vuelve al proyecto “no revisable” y recibe un 0 de calificación inmediata:

- Todo el código debe estar escrito en C
- El proyecto debe compilar y ejecutar en Linux (*i.e.*, debe estar “integrado”, explicaciones del tipo “*todo está bien pero no pusimos pegarlo*” provocan la cancelación automática de la revisión).
- El proyecto se debe presentar en una **máquina física** que levante en Linux (puede ser dual)

VIII. FECHA DE ENTREGA

Demostraciones el **Martes 14 de Noviembre - 9:30am**. Mande además un `.tgz` con todo lo necesario (fuentes, `makefile`, `readme`, etc.) a `torresrojas.cursos@gmail.com`. Ponga como subject: `COMPILADORES - Proyecto 3 - Fulano - Mengano - etc.`, donde Fulano, Mengano y etc. son los miembros del grupo.