

Diseño y Construcción de Compiladores - 2023

Práctico de Máquina: Guía general

El práctico de máquina a desarrollar en esta materia consiste en el diseño y construcción de un **compilador para un lenguaje reducido de C++**.

1. Cronograma

El práctico de máquina consta de 3 etapas bien diferenciadas. Cada una de ellas cuenta con una corrección y su recuperación en las fechas descriptas en el cronograma de la materia. Al aprobar la segunda entrega, se deberá presentar un **informe** en el cual estén detalladas todas las decisiones de diseño que se han tomado y su justificación, como así también una breve descripción de cómo ha sido la experiencia de trabajar en grupo y afrontar este desafío. La aprobación de las tres entregas más la entrega del informe es necesario como parte de la regularización de la materia.

2. Metodología de trabajo

Los códigos serán cargados en la plataforma en las fechas estipuladas como un archivo comprimido. En el mismo se debe incluir el archivo **ucc** compilado, los códigos fuentes y *una descripción puntual de las decisiones que se han tomado* para abordar la etapa. El compilador será probado con lotes generales y luego, con lotes especializados de acuerdo a la descripción brindada con cada entrega.

3. Instrucciones de uso

- **Forma de compilar el compilador ucc**

- Para compilar y generar el ejecutable **ucc** simplemente debe abrir una ventana con la consola, se debe posicionar en la carpeta en la que están los archivos fuentes del compilador que está construyendo y debe escribir: **make**
- Para eliminar los archivos objeto generados y el **ucc** puede utilizar: **make clean**

- **Cómo usar el compilador ucc**

El archivo ejecutable **ucc** respeta el siguiente formato:

- Para compilar un archivo fuente **.c** escrito en nuestro sublenguaje:
./ucc -c lote_de_prueba.c
- Para ejecutar un archivo objeto **.o** generado por **ucc** a partir de una compilación exitosa:
./ucc -o lote_de_prueba.o

4. Módulos del Compilador

El programa entregado como parte del práctico de máquina se compone de varios módulos, los cuales se irán entregando durante las distintas etapas y se describen a continuación.

- **Módulo Parser**

Este módulo se compone de **parser.c** y su header **parser.h** quien a su vez incluye otros headers; por ejemplo **var_globales.h**. El archivo **parser.c** es la implementación del parser descendente recursivo (PDR) para el subconjunto de la gramática de C++. El mismo es el eje central del compilador, dado que es el encargado de establecer el orden de las invocaciones al resto de los módulos. El archivo **var_globales.h**, como su nombre lo indica, tiene las declaraciones de las variables globales.

- **Módulo Scanner**

Este módulo incluye el archivo **lex.yy.c**, el cual se obtiene a partir del archivo **scanner.token.lex** al compilarlo con **lex**. El **parser.c** se encarga de llamar a la función **yylex()**, a través de la función **scanner()** definida en él. Además, está el archivo **codigos.h** que es el que tiene los códigos devueltos por el scanner definidos como constantes.

- **Módulo Errores**

El módulo de errores se compone del archivo **error.c** y de **error.h**. A su vez, **error.c** posee dos funciones que son invocadas desde **parser.c**, **lex.yy.c** y **ts.c**:

- **void error_handler(int ne)**

Función que maneja la **salida del compilador**. Cuando la función recibe como argumento **COD_IMP_ERRORES** imprimirá la línea actual y sucesivamente los errores correspondientes. Si un código de error es pasado como argumento, éste es agregado a la estructura definida en **error.h**, la cual va soportando los errores para la línea corriente. **error_handler(COD_IMP_ERRORES)** es invocada principalmente desde **lex.yy.c** cuando reconoce el caracter *new line*.

- **void error_print(int ne)**

Función que contiene los posibles mensajes de errores en la etapa de compilación.

- **Módulo Tabla de Símbolos**

Este módulo contiene las funciones necesarias para la administración de la Tabla de Símbolos. Dichas funciones son las encargadas de manejar los objetos de datos definidos o declarados en el archivo fuente a compilar. El módulo de la Tabla de Símbolos está compuesto de un archivo header **ts.h** y un archivo **ts.c**.

- **Módulo Sistema de Ejecución**

Este módulo mantiene las estructuras soportes para el tiempo de ejecución e implementa las instrucciones de la **Máquina Abstracta C**. Está conformado por los archivos **sistejec.h** y **sistejec.c**.