

Práctico de Máquina - Tercera Entrega
Generación de Código Intermedio y Máquina Abstracta
Diseño y Construcción de Compiladores - 2022

En esta tercera etapa, el compilador deberá generar **Código Intermedio** para que la **Máquina Abstracta** pueda ejecutarlo.

1. Generación de Código Intermedio

Se deberá generar el código, en un archivo `.o`, al menos para las siguientes construcciones:

- Declaración de variables simples.
- Proposición de selección.
- Proposición de iteración.
- Proposición de entrada-salida.
- Expresiones en general.

Nota: Cada estudiante deberá decidir si genera código para la proposición de selección o bien para la proposición de iteración.

2. Máquina Abstracta C

La cátedra brinda el sistema de ejecución implementado, es decir, la máquina abstracta C. En esta etapa se debe entregar el compilador funcionando de la siguiente manera:

- Si el programa a compilar tiene errores, los mismos deben ser mostrados y no se debe generar código.
- Si el programa a compilar es correcto, el comando `ucc -c programa.c` debe generar código en el archivo `programa.o`. Luego, con el comando `ucc -o programa.o` se ejecutará el programa.

3. Recomendaciones Generales

- Utilice en su código parser, una variable de las mismas características que la memoria de programa **P** de la máquina abstracta C, donde ir almacenando el código generado. Si por ejemplo esa variable se llama **CODE**, antes de comenzar a generar el código del programa propiamente dicho, **CODE[0]** debe contener la instrucción **INPP** y en su última posición debe contener **PARAR**. Entre estas dos instrucciones de la MAC se debe encontrar todo el programa generado. En el caso de que el programa a compilar tuviese constantes de tipo string, éstas también deben ser almacenadas en otra estructura auxiliar de las mismas características que la zona de constantes string **C** de la MAC.
- Para generar el archivo objeto **programa.o** utilice la función **guardar_codgen** provista en el archivo "sistejec.c".