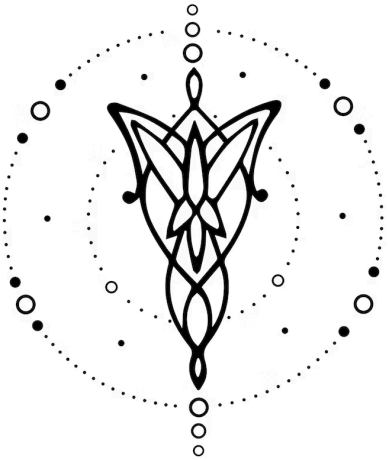

Especificación

Diseño e Implementación de un Lenguaje

v1.0.0

1. Equipo

Nombre	Apellido	Legajo	E-mail
Valentino	Sanguinetti	63098	vsanguinetti@itba.edu.ar
Luciano	Stupnik	64233	lstupnik@itba.edu.ar
Matias Juan	Rossi Seifert	63202	mrossiseifert@itba.edu.ar
Junior	Rambau	64461	jrambau@itba.edu.ar

2. Repositorio

La solución y documentación serán versionadas en:

https://github.com/valensangui8/microGCC

3. Dominio

Se desarrollará un lenguaje simplificado derivado del lenguaje C, que permita traducir un subconjunto básico de instrucciones y estructuras a código Assembly (ASM x86-64, sintaxis intel). Este lenguaje se centrará en construcciones fundamentales de programación imperativa, como asignaciones, estructuras condicionales, bucles, definición básica de funciones, declaración de variables y operaciones aritméticas y lógicas.

La implementación del lenguaje permitirá a los desarrolladores familiarizarse con los conceptos fundamentales de la compilación, manejo de registros, pilas, control de flujo y gestión de memoria, brindando una herramienta educativa y práctica para experimentar con la traducción directa entre código de alto nivel y lenguaje máquina.

4. Construcciones

El lenguaje propuesto contará con las siguientes construcciones, prestaciones y funcionalidades:

- Declaración y asignación de variables (int, char).
- Operaciones aritméticas básicas: suma (+), resta (-), multiplicación (*), división (/).
- Operaciones lógicas: AND (&&), OR (||), NOT (!).
- Operadores relacionales: igual (==), distinto (!=), mayor (>), menor (<), mayor o igual (>=), menor o igual (<=).
- Estructuras condicionales simples y compuestas (`if`, `if-else`).
- Estructuras repetitivas básicas: `while`, `for`.
- Definición y llamada a funciones con parámetros y retorno de valores.
- Retorno de valores desde funciones con la palabra clave `return`.
- Declaración de una función externa utilizando la palabra clave `extern` para poder tener llamados a funciones de otros archivos en el archivo generado.

5. Casos de Prueba

Casos de aceptación

1. Programa que declara variables enteras y asigna valores.
2. Programa que realiza una suma simple de dos variables.
3. Programa que usa condicionales `if-else`.
4. Programa que implementa un bucle `while` sencillo.
5. Programa que utiliza la estructura de control `for`.
6. Programa que define una función simple sin parámetros.
7. Programa que define y llama una función con parámetros enteros.
8. Programa que realiza operaciones lógicas.
9. Programa que realiza operaciones relacionales.
10. Programa que combina asignaciones, condicionales y ciclos en una función.
11. Programa que hace un llamado a una función externa.

Casos de rechazo

1. Programa con declaración mal formada de variables (tipo de datos no existente o sintaxis incorrecta).
2. Uso de variables no declaradas previamente.
3. Intento de asignar a una variable un valor de otra variable de tipo distinto.
4. Sintaxis incorrecta en estructura condicional.
5. Intento de realizar operaciones entre tipos incompatibles (char e int).
6. Definición de funciones sin especificar tipo de retorno.

6. Ejemplos

Ejemplo 1: Declaración y suma simple

```
int main() {
    int a;
    int b;
    int c;

    a = 5;
    b = 10;
    c = a + b;

    return c;
}
```

Ejemplo 2: Condicional simple

```
int check(int num) {
    if (num > 0) {
        return 1;
    } else {
        return 0;
    }
}
```