

CompilerCRB

Overview

CompilerCRB is a lightweight compiler written in C for compiling a subset of the C# programming language. Designed for educational purposes, this compiler demonstrates the fundamental concepts of compiler design, including:

- Lexical Analysis
- Parsing
- Semantic Analysis
- Intermediate Representation (IR)
- Code Generation

Features

- Supports basic data types: integers
- Handles variable declarations and assignments
- Implements simple arithmetic operations (+ , - , * , /)
- Constructs an Abstract Syntax Tree (AST)
- Performs semantic analysis with a symbol table
- Generates a simple assembly-like target code

Getting Started

Prerequisites

- GCC or Clang compiler
- Make

Building the Compiler

1. Clone the Repository:

```
git clone https://github.com/DavidVardzelian/CompilerCRB
cd CompilerCRB
```

2. Build the Project:

```
make
```

This will compile the source files and generate the executable `CompilerCRB` in the `bin/` directory.

Running the Compiler

1. Create a Source File:

Create a simple source file, e.g., `test1.src` in the `tests/` directory:

```
x = 5 + 3;
y = x * 2;
```

2. Compile the Source File:

```
./bin/CompilerCRB tests/test1.src > output.asm
```

3. View the Generated Assembly:

```
cat output.asm
```

Sample Output:

```
MOV t12345, 5
MOV x, t12345
MOV t67890, 3
ADD t12345, t67890
MOV x, t12345
MOV t54321, x
MOV t09876, 2
MUL t54321, t09876
MOV y, ACC
```

