Cairo University

Faculty of Engineering Credit Hours System

Languages and Compilers

New C Compiler

Submitted by:

Hla Hany Mohamed Helmy	1190344
Mostafa Osama AbdelZaher	1190173
Omar Mohamed Ahmed	1190204
Yomna Osama Hussien	1190203

Overview:

The project is a modified version of the C compiler, it take input as from a text file or the GUI and outputs the symbol table and the quadruples. It also has error handling and will show if any syntax error occurred.

Tools and Technologies used:

We used YACC and Lexer to do the lexical analysis and parsing, for the logic used to implement the tables and quadruples we used C++, we used Make and g++ to compile the C++ files with the lex and YACC, we also have a docker container that has all the required dependencies. For the GUI we used PyQt.

Tokens:

INTGER_NUMBER	Integer numbers (0, 1 , 2 etc...)
FLOAT_NUMBER	Floats (1.2,8.4 etc..)
STRING_IDENTIFIER	“any string”
CHAR_IDENTIFIER	‘c’
TRUEE/ FALSEE	TRUE/FALSE
INT FLOAT CHAR STRING BOOL VOID	Int, float, char, string, bool, void (data types)
NULLL CONST	null and const
INCREMENT DECREMENT	++ and --
GREATERTHANEQUAL LESSTHANEQUAL GREATERTHAN LESSTHAN NOTEQUAL EQUAL	>=, <=, >, <, !=, == (comparators)
AND OR NOT	&&, ||, !
IF ELSE WHILE FOR DO BREAK CONTINUE RETURN SWITCH CASE DEFAULT	if, else, while, for, do while, break, continue, return, switch(), case x:, default:
IDENTIFIER	Any identifier string; x, num, val etc...

Quadruples:

MOV a x	x = a
ADD a b t1 SUB a b t1 MUL a b t1 DIV a b t1 MOD a b t1	t1 = a + b t1 = a – b t1 = a*b t1 = a/b t1 = a%b
CMP a b	Compares 2 values, then we use conditional jump after
JLE label, JGE label, JNE label, JE label, JLT label, JGT label	Jump less than or equal, Jump Greater than or Equal, Jump not Equal, Jump Equal, Jump Less than, Jump Greater than, after a comparison to a label
Label:	A label to jump to
Call function	Goes to the function name

JMP label	Unconditional jump to line
INC x, DEC x	++x, --x

Sample:

Input:

```
int x = 10;
int y = 20;
if (x > 10) {
    y = 30;
}
else{
    if (x < 10) {
        y = 40;
    }
    else{
        y = 50;
    }
}
```

Output:

```
MOV 10  x
 MOV 20  y
 CMP x 10
 JLE   Line0
 MOV 30  y
 JMP   Line1
 Line0:
 CMP x 10
 JGE   Line2
 MOV 40  y
 JMP   Line3
 Line2:
 MOV 50  y
 Line3:
 Line1:
```