CMPE230 Homework 1

Burak Yüksel – Fahri Can Şanlı

2015400225 - 2015400096

Problem Statement

We are requested to write a C++ code that converts a comp file, which consists of expressions and assignment statements, to an assembly code.

Solution Description

First, we removed whitespaces from input. Then we converted expressions which is given in infix notation to postfix notation by using the method we covered in the lecture. After obtaining postfix version of input expressions, we started iterating through each line and we produced corresponding assembly code.

At this stage, since we cannot store 32 bit values in one register, we divided all variables and numbers into two parts. We also created new variables to use in multiplication and power operations. Besides, we renamed all variables by using map to prevent the case sensitivity problem.

While producing assembly code, we also checked if there are any errors in expression

Functions

read (): We implemented this function to read input file.

edit (): We implemented this function to delete whitespaces from each line and also to check if there is a parenthesis error.

expr (): We implemented this function in the process of converting to postfix.

term (): We implemented this function in the process of converting to postfix.

factor (): We implemented this function to check if the next statement is an expression with parentheses or a number or a variable or a power operation.

morefactors (): We implemented this function if there is a multiplication operator in the expression.

moretems (): We implemented this function if there is a addition operator in the expression.

to\_postfix (): We implemented this function to convert infix notation to postfix notation.

to\_assembly (): We implemented this function to produce a corresponding assembly code.

multiply (): We implemented this function to overcome multiple use of multiplication process.

How to Compile

1. Compile C++ source code: g++ code.cpp -o comp
2. Running C++ code to get assembly code: ./comp [file\_name].co
3. Running assembly code: a86 [file\_name].asm