Muhammed Emin Ayar- Hatice Melike Ecevit
2015400216-2016400138
25 March 2018

CMPE230 2018 Spring Project1

Problem Description

In this project, we are expected to implement a compiler which takes a file consisting some expression and assignment statements and generates the corresponding a86 code.

Assumptions and Constraints

1. A single statement cannot overlap a single line.
2. A single statement cannot have more than one equal sign.
3. Variables cannot have the name ‘pow’.
4. Variable names cannot contain spaces.

Implementation

In our solution we process the file 2 times. In the first parsing we check if all statements are legal and take all variables in the file and declare them in our assembly code. In the second parsing we process the statements line by line. First we check for global plusses which are plusses that are not in parenthesis or are not in pow operation(parseThis). We split it in terms of that and send it to another function(pushToStack) which does the same for multiplication operation. These functions work recursively and call each other for processing the statement. For our assembly implementation we wrote 4 functions which are doPower, doMult, Printer, and doSum. The function doPower pops two elements from the stack and does the pow operation in O(log(b)) (for pow(a,b)) in assembly. The function doMult pops two elements from the stack and multiplies them in assembly. The function Printer pops an element from the stack and prints it in hexadecimal. The function doSum pops two elements from the stack and add them in assembly. The function isIllegal checks for valid parentheses, multiple plusses or multiplication signs side by side, open parentheses and a plus sign or multiplication sign side by side, correct usage of pow operation.

Conclusion and Evaluation

We are not expected to check every syntax error but we have checked most of them. We wrote our code in a way that assembly does the whole computation and memory access. In our C++ code we just parse the file and process the statements.

How to Compile & Run

**add the following line to your .bashrc file
alias comp='./parser'

Compile command:
g++ code.cpp -o2 -w -std=c++11 -o parser

Run command:
comp example.co