CMPE 230: SYSTEMS PROGRAMMING

01.05.2023

PROJECT 2

PREPARED BY:

* ERAY EROĞLU 2020400096
* BERKE KARTAL 2020400198

INTRODUCTION

The project that we are going to discuss is a transcompiler, a simple program which generates an LLVM IR format assembly code. The program reads a file, which is consisted of arithmetic operations, and writes the corresponding output to a file. Source code is written in C language.

The source code is implemented upon the previous project’s source code. In the previous project, the main algorithm was creating a parsing tree corresponding the current arithmetic operation. This is done by several parsing methods and the result is calculated by an evaluating method.

These parsing methods and the evaluating method are also used in the current project, with slight changes. The main idea behind the parsing tree is still same, the given expression is parsed into terms and factors. After creating the tree, evaluating methods calculates the result by traversing the tree recursively. For further details about these methods, previous project’s documentation can be examined.

INPUT/OUTPUT

As a difference from the previous project, the transcompiler doesn’t take inputs from terminal as arithmetic operations. These operations should be stored in a file, and the program takes the path of this file as an input. The corresponding output is written to another file with the same name but “.ll” extension.

However, this output file is generated as long as all the arithmetic operations in the input file is valid. There is an additional invalid input case, comparing to the previous project, which is the use of undefined variables. If the user tries to use a variable, which wasn’t assigned any value yet, it is evaluated as error. For other invalid input cases, previous projects documentation can be read.