CS4318: Compiler Constructions
Project Phase 4: Code Generator for mC
100 Points

Submission Deadline: 4th May, Sunday. 11:59 PM

Objectives

Write a code generator for mC

Description

In this final phase of your project you will add a code generator module to your compiler. You are free to target your compiler for any architecture. Whatever platform you choose, we should be able to assemble your code with the native assembler and link it to an executable. The resulting code should execute without errors.

The recommended target is MIPS. If you choose MIPS as your target, you can test your programs with the SPIM simulator. Reference manuals describing the calling convention and assembly instructions can be found as appendices in the Patterson and Hennessey architecture text (Computer Organization and Design, The Hardware/Software Interface, 5th Edition).

Your compiler should be able to generate code for the following language constructs in mC :

1. expressions
2. assignment statements
3. conditional statements
4. iterative statements

The execution semantics for these are the same as that of C.

Register Allocation

You do not have to implement the global graph coloring register allocator (although it would be nice, if you did). A simple scheme where you save all registers before function body execution and reuse registers that hold live values will be sufficient. Also, you are allowed to spill registers to memory when needed.

Implementation and Testing Instructions

- Your program should be compiled using the commands **`make clean; make`**.
- During the submission, please make sure that the latest versions of all of the following files are present in submitted folder – **`driver.c, scanner.l, parser.y, strtab.c, strtab.h, tree.h, tree.c, codegen.c, makefile, and writeup.txt`**.
- Note that only **`codegen.c`** file is new. You should already have the other files from your previous submissions. It is okay if you make corrections to the other files. Make sure to check in the latest code.

- Edit all the files inside the folder named ”`Generated_Codes`”. Edit the text files and provide the outputs that your code generates for the corresponding inputs in the “`Sample_Test`” folder.
- To avoid any confusions unlike the previous phases, be sure to run your code in zeus server and attach screenshots for all the inputs given in the folder.

Writeup

You need to write a short document named `writeup.txt`.
In it, you should include the following (numbered) sections.

1. Your name, course id.
2. If a group project, then who contributed in which part.
3. Extra grading instructions if you choose not to use makefile.

Compilation, Testing, and Submission

- Your program should be compiled using the commands `make clean; make.`
- You are allowed to do otherwise but make sure to add proper instructions in the `writeup.txt` file.
- The makefile for building your scanner is included.
- If you have made corrections to your previous files, make sure to include those.

Table 1: Rubric for Project Phase 2

Category	Points
Expressions	30
Assignment Statements	15
Conditional Statements	15
Iterative Statements	15
Undeclared and multiply declared variables	5
Documentation	10
Write up	10
Compile & Run	10

Submission Deadline: 4th May, Sunday. 11:59 PM
(This deadline is final and there will be no extension)