Assignment 4

March 17, 2018

Deadline

11:55 PM on 1st April, 2018

Problem

Write an ***Intermediate Code Generator*** for the intermediate representation that you designed in Assignment 2. The output of the assignment must be a ***linked-list of Three-Address Code (3AC) statements*** corresponding to each function in the input program. This assignment requires you to:

- attach semantic actions to the parser implemented in Assignment 3 to:
  - generate intermediate representation as a linked-list of three-address code statements;
  - the associated symbol table.
- print the three-address code on the standard output.

It should support all types supported by the subset of the source language identified by you (including arrays and pointers). It should also have support for function calls.

Details

- Your implementation should read the source filename as its first command-line parameter; it should produce its output on the standard output as a listing of three-address code statements.
- You must use **your** implementation of the lexer and parser from the prior submissions. You are allowed only minor modifications over your Assignment 3 submission.
- Your intermediate representation must adhere to your design of the representation that you used in Assignment 2 (minor modifications allowed).
- You must use a parser generator like Yacc, Bison etc.
- The tool should be robust; syntactically incorrect programs must be duly reported.
- You have to submit a zipped folder (name the folder "asgn4") with:
  - the source of the implementation (in a folder called "src" within "asgn4";
  - a Makefile to build the implementation (it should generate an executable called "irgen" in the folder "asgn4/bin";
  - a set of at least 5 test cases that you have used to check your implementation (in a folder "asgn4/test");
  - a README file with a brief description for building and running it (within "asgn4").

  Binaries should NOT be part of the submission. Clean the folder of all object and executable files before submission.
- Note that all elements of your submission, like code quality and readability, quality of documentation (README file), quality of test-cases etc. carry marks.
- We will apply the following set of commands to build and run your implementation; make sure that your implementation works correctly with these sequence of commands:

- cd asgn4
- make
- bin/irgen test/test1.c (to execute the first test-case file test1.c; the sequence of three-address code statements should be displayed on the standard output)