



Assignment 5

Please submit solutions on Blackboard by Monday, 24.03.2020 23:59h

5.1 Practical Assignment: Symbol Table Construction (Points = 125/200)

5.1.1 Part I = 50 points

The VSL compiler in the provided archive is extended with a function 'generate program' in generator.c; this function is called from main.c, after syntax tree and symbol table construction. Implement this function so that it generates x86_64 assembly code for the following constructs:

- **Global String table:** Strings should be given numbered labels in a data segment.
- **Global variables:** Global variables should be given names corresponding to their declarations, prefixed with an underscore character '_', so as to avoid names that clash with names from the system libraries.
- **Functions:** Functions should be placed in the text segment, named in the same manner as global variables, and set up/remove a stack frame. Furthermore, they should initiate a recursive traversal of their syntax subtrees, so that the remaining constructs can be generated.
- **Function Parameters:** Function parameters should be expected to follow the standard calling convention covered in lectures. Copies can be placed at the bottom of the function's stack frame, to make their run-time address computable from their sequence number, and liberate the registers for further function calls.
- **Arithmetic expressions:** Arithmetic expressions should be translated so as to leave their result in the RAX register, and remove any intermediate calculations from the generated program's run time stack.
- **Assignment statements:** Assignment statements should copy the result of an expression to the address of the assigned variable.
- **print statements:** "print" statements can be translated into a sequence of 'printf' calls, with one call per item in the PRINT statement's list.
- **return statements** "return" statements should leave the result of their expression in the RAX register, remove the function's stack frame, and return control to the caller.

5.1.2 Part II = 75 points

Implement the following constructs in generator.c:

- a. Local variables 20%
- b. Function calls 20%
- c. Conditionals (IF and relations) 15%
- d. While loops 10%
- e. Continue (null statement) 5%