Project 4

Small C compiler for x86

The goal of our project this semester is to develop a prototype C compiler. This is done by compiling C programs into embedded platforms such as 68K, MIPS, ARM, or pseudo assemblies. In this class we will mainly use x86 as the target codes. The project is divided into several parts including language definition, scanner, C-grammar, symbol table handlings, parser, type checker, and code generation. In the final assignment, you will need to choose the set of language features which you want to develop the compiler for in this course, and write the whole compiler (including the code generator). The generated x86 assembly code is expected to be executed on the machines, mcore8.cs.ccu.edu.tw or linux.cs.ccu.edu.tw.

In your hand-in report, you need to have the followings:

- **To define the subset of the language which you want to choose from C.**
- Give a set of testing programs which can illustrate the features of your testing programs. (at least 3 test programs)
- Use the "**ANTLR**" to help you develop your compiler.
- You can use **C** or **Java** to write your compiler. (Java is recommended)
- Please ensure your program can be executed under the **mcore8** or **linux.cs.ccu.edu.tw** workstations.
- Support (at least two parameters) *printf* function in your compiler.
  - Ex: `printf("%d\n", var);`
- You can assume the compiler can always get a free register to use.

Please turn in the following files to ECOURSE:

- A file describes your language which is a subset of C language. (MS-WORD file)
- The source codes:
  - Your compiler
    (a) ANTLR grammar file, myCompiler.g.
    (b) A program to call your parser, myCompiler_test.
  - Testing programs (.c) and the corresponding x86 assembly codes (.s) generated by your compiler. (at least 3 programs)
- A readme file (pure text file) describes the features of your compiler and explains how to compile and execute your compiler.
- Makefile

Due Date: June 29 (Friday), 24:00, 2018