Project 1

Lexical Analyzer

The goal of our project this semester is to develop a prototype C compiler. This is done by compiling C programs into processors such as x86, MIPS, ARM, or pseudo assembly codes. In this class, we will use LLVM IR as the target code. **The project is divided into several parts including language definition, lexical analyzer, C-grammar, symbol table handlings, parser, and code generation.** In the first part, you will need to choose the set of language features you want to support in your compiler, and write the lexical analyzer.

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 that can illustrate the features of your C compiler. (at least 3 test programs)
- Use the "**ANTLR**" to help you develop the lexical analyzer.
- You can use **C, C++**, **Java**, or other programming languages to write your lexical analyzer. (**Java is recommended**)
- Output the token-type and content of each token.

```
Token:1  int
Token:4  <=
…
```

- Please ensure your program can be correctly executed under the **tux.cs.ccu.edu.tw** workstation. (optional)

Please turn in the following:

- A file describes your language, a subset of C language. (MS-WORD file)
- The source codes:
  - ANTLR grammar file, `mylexer.g`.
  - A program to call your lexer, `testLexer.java`.
  - Testing programs. (at least 3 programs)
- A readme file (pure text file) describes how to compile and execute your lexical analyzer.
- A "Makefile" file.

請將上述的檔案打包壓縮存成 tar.xz 之格式，檔名為<學號>.tar.xz 檔 (例如: 401234567.tar.xz，上傳到 eCourse2。

Due Date: March 27 (Wednesday), 24:00pm, 2024