Assignment 1

January 8, 2016

Problem

Write a Lexer for the source language that you have chosen. The output of the lexer must be a "summary" of the tokens in the program.

Example

For a program in the C language:

Input:

```
main()
{
        int a = 959;
        foo(a);
        return 0;
}
```

Expected Output:

Token	Occurrances	Lexemes
OP_ASGN	1	=
'('	2	(
')'	2	)
IDENTIFIER	3	a
		foo
		main
BLOCK_BEGIN	1	{
BLOCK_END	1	}
TYPE	1	int
KEYWORD_RET	1	return
INT_CONST	2	959
		1
STMT_TERMINATOR	3	;

Details

- You are free to select your own token names.
- Your implementation should read the source filename as its first command-line parameter; it should produce its output on STDOUT.
- You must use a lexer generator like Lex, Flex etc.
- The tool should be robust; any failure in tokenizing due to errors in the input program must be reported properly.

- You have to submit a zipped folder (name the folder "asgn1") with:
  - the source of the implementation (in a folder called "src" within "asgn1";
  - a Makefile to build the implementation (it should generate an executable called "lexer" in the folder "asgn1/bin";
  - a set of at least 5 test cases that you have used to check your implementation (in a folder "asgn1/test");
  - a README file with a brief description for building and running it (within "asgn1").

  Binaries should NOT be part of the submission. Clean the folder of all object and executable files before submission.
- 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 asgn1
  - make
  - bin/lexer test/test1.c (to execute the first test-case file test1.c)