Project1

1. Introduction

In this experiment, we used Flex and Bison to achieve simple lexical and syntactic analysis. The basic function is to enter a copy of the SPL source code and detect if there are lexical and syntax errors, and if there are errors, the type of typo, and the line number. If there are no errors, print the syntax tree.

2. Basic Function

1. Identify lexical errors: Any undefined character and any character that does not conform to the definition of the lexical unit occurs
2. Identify syntax errors
3. Recognize hexadecimal digits and characters
4. Print the syntax tree

3. Compilation Step and Usage

```
splc:

	@rm  -rf  bin/

	@rm  -rf  lex.yy.*  syntax.tab.*

	@mkdir  bin

	@touch  bin/splc

	@chmod  +x  bin/splc

	@flex  lex.l  &&  bison  -t  -d  syntax.y  &&  gcc  main.c  tree.c  syntax.tab.c  -lfl  -ly  -o  ./bin/splc

clean:

	@rm  -rf  bin/
```

```
        @rm -rf lex.yy.* syntax.tab.*

.PHONY: splc
```

These code can be found in Makefile.

To use this project, first ***make***, then use ***/bin/splc*** to process the file you want.

Program Structure

1. main.c
   - Program entry, read in the file, call bISON internal function for syntax analysis
2. tree.h with the tree.c
   - Defined node types of tree and define the function to insert node and the print syntax tree.
3. lex.l
   - Conduct lexical analysis
4. syntax.y
   - Perform grammatical analysis

Lexical Analysis

Basically, the match is performed using the identifiers given by the documentation, and the unspecified identifiers are uniformly specified as a token called LEX_ERROR to avoid undesirable results from parsing

Syntax Analysis

Syntax parsing basically refers to the expressions in the given document, adding error handling where possible to print syntax errors, and it should be noted that the relationship between nodes needs to be handled during the matching process.

Syntax Tree

The syntax tree is realized by a simple multi-tree, which implements the insertion method and prints the syntax tree through a simple preorder traversal.

Test

I passed the 12 tests given and 5 self-written tests.