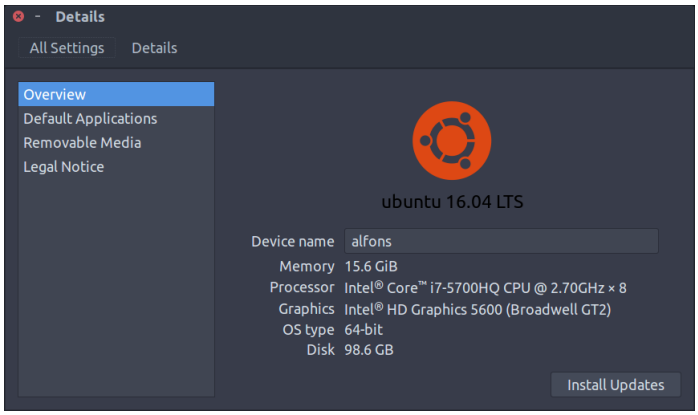
Introduction to Compiler Design Project1 Report

0416324

1.Programming platform:

Ubuntu 16.04 x64 with lex, gcc and c language.



2.Lex scanner ability

Able to detect the necessary elements including.

(1) Delimiters

(2) Arithmetic, Relational, and Logical Operators

(3) Keywords

(4) Identifiers

(5) Integer Constants

(6) Scientific Notations

(7) Floating-point Constants

(8) String Constants

(9) Control character [\n]

Able to discard the unnecessary elements, but still to the necessary processing

(1) Whitespace and tab[\r]

Use the LIST function in lex template to do the source code indentation

correctly

(2) Comments

Detecting the following two types of comments and specify the illegal comments.

e.g. /*This is a comment */ is a legal c style comment

/* this is a comment // line with some /* and

// delimiters */ is also a legal c style comment

However, /*this is a tricky comment*/ this part is not considered to be a comment but some identifiers*/ is not a legal comment (the part between */ and */ is not the part of c style comment

// this is a comment // line */ /* with some /* delimiters */ before the end is a legal c style comments

(3) Pseudocomments

Detect and use them to activate/deactive the function in lex scanner.

I use the following tricky testcase to test my scanner.

Testcase is at: https://pastebin.com/83zRkxmB

Output result is at: https://pastebin.com/ujCEJ0CB

(The lextemplate has been modified to fully detect all the bad characters, which will loop till the end to detect any illegal characters not specified in the project requirements and specify the C and C++ style comments respectively)

Lextemplate is able to specify the hardest part: delimiting two comments /*c1*/ /*c2*/ modified from the regex of only [/][.](.|[\n])[/][*] which will specify only one comment since lex uses greedy method(or say greedy algorithm), the modified regex should be

[/][*]([^*]|[\n]|[*]+[^/]|[/][*])*[*]+[/] which means

(1) We may not permit the star symbol in the middle (or we may, but the star symbol CANNOT BE FOLLOWED BY THE SLASH since the slash symbol only appears at the end of the comments.

(2) The \n new line symbol is legal in the c style comment

(3) If the star symbol has to be appeared in the comment, then there should be no / (slash symbol) after it, since */ should only appear in the end of the comment.

(4) The /* in the symbol is acceptable just as the problem defined in the project pdf

Also it can detect the wrong style of comment such as /**/xxx */ the xxx part should be specified as identifier, delimiter or relational, arithmetical operator.

(5) [*]+ at the end before / to accept multi star symbol before the slash such as

/*wfiuhwef/*piuhwefiu*****/ is still a legal C style comment.

For the other illegal elements that need to be specified out and terminated.

Such as 9”9 or ”999 ”9”9””9 and 9999&@#@!$#@(these symbols are not considered to be the delimiters), lots of trash like them should be detected and generate error message.

Use regex ["]?([0-9]+["])+ once a ” append with integer or a integer cross-appearing with an ” are both illegal. (consider the ”999” is still legal since it is a string)

Or such identifier like 9ast 88you 100_int is not legal.

Use regex [0-9]+[a-zA-Z_] to identify an illegal identifier

And the rest like *&ˆ(*&ˆ trash should be treated as the lowest precedence and use . (dot symbol or say the rest) after all the rule are exhausted.

For pesudocomments , they all have the same form

"//&S-"(.)*[\n]

and once encountered, do the necessary modification of option and line_num++ to count it as a whole line.

3. How to build and run the program

Using the makefile in Linux and aggregate all the necessary compilation altogether.

(1) Use lex to generate lex.yy.c from the source code of lextemplate

$ lex lextemplate.l

(2) Gcc it, -lfl tells gcc to use the lex(or flex) library

$ gcc -o scanner lex.yy.c -lfl

(3) Run it (file in the same path folder)

$ ./scanner [filename]