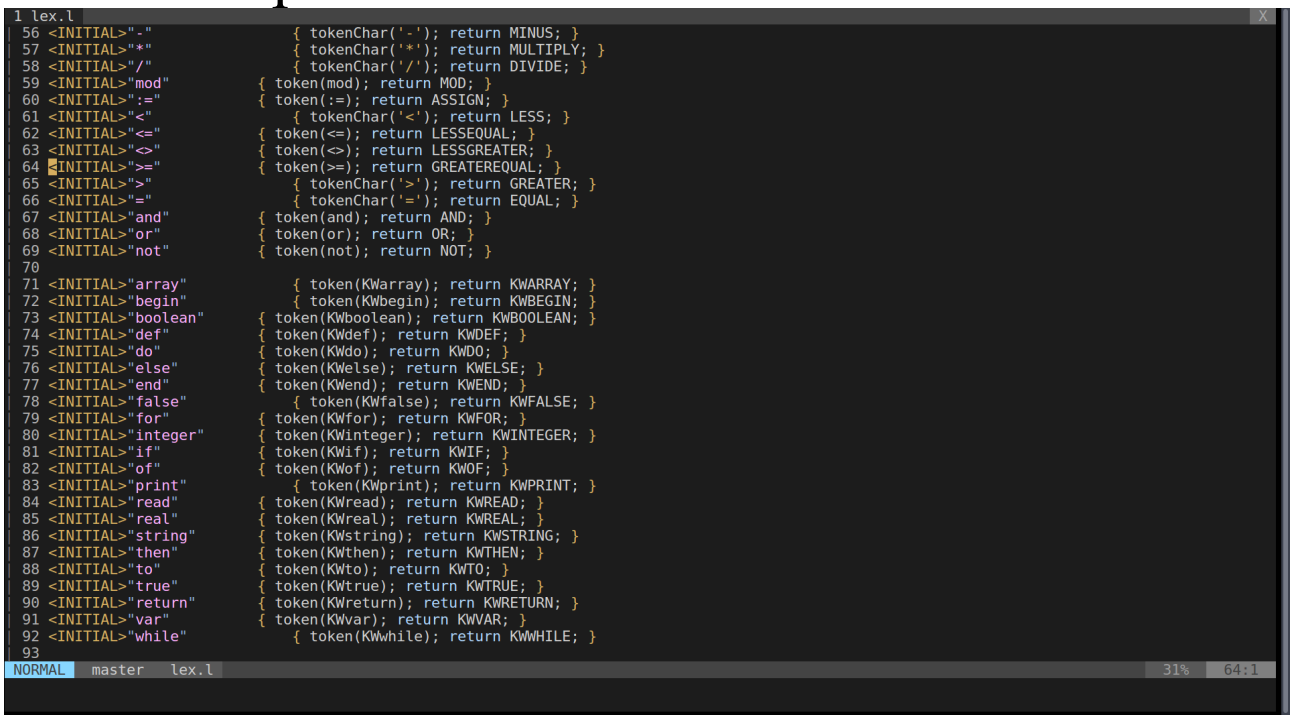
Intro. To Compiler Design Proj2 Report

0416324 An – Fong Hwu

1.Part of Lex I have modified to make it interact with Yacc parser?



```
1 lex.l
 56 <INITIAL>"-"              { tokenChar('-'); return MINUS; }
 57 <INITIAL>"*"              { tokenChar('*'); return MULTIPLY; }
 58 <INITIAL>"/"              { tokenChar('/'); return DIVIDE; }
 59 <INITIAL>"mod"          { token(mod); return MOD; }
 60 <INITIAL>":="           { token(:=); return ASSIGN; }
 61 <INITIAL>"<"              { tokenChar('<'); return LESS; }
 62 <INITIAL>"<="           { token(<=); return LESSEQUAL; }
 63 <INITIAL>"<>"           { token(<>); return LESSGREATER; }
 64 <INITIAL>">="           { token(>=); return GREATEREQUAL; }
 65 <INITIAL>">"              { tokenChar('>'); return GREATER; }
 66 <INITIAL>"="              { tokenChar('='); return EQUAL; }
 67 <INITIAL>"and"          { token(and); return AND; }
 68 <INITIAL>"or"           { token(or); return OR; }
 69 <INITIAL>"not"          { token(not); return NOT; }
 70
 71 <INITIAL>"array"            { token(KWarray); return KWARRAY; }
 72 <INITIAL>"begin"            { token(KWbegin); return KWBEGIN; }
 73 <INITIAL>"boolean"      { token(KWboolean); return KWBOOLEAN; }
 74 <INITIAL>"def"          { token(KWdef); return KWDEF; }
 75 <INITIAL>"do"           { token(KWdo); return KWDO; }
 76 <INITIAL>"else"         { token(KWelse); return KWELSE; }
 77 <INITIAL>"end"          { token(KWend); return KWEND; }
 78 <INITIAL>"false"            { token(KWfalse); return KWFALSE; }
 79 <INITIAL>"for"          { token(KWfor); return KWFOR; }
 80 <INITIAL>"integer"      { token(KWinteger); return KWINTEGER; }
 81 <INITIAL>"if"           { token(KWif); return KWIF; }
 82 <INITIAL>"of"           { token(KWof); return KWOF; }
 83 <INITIAL>"print"            { token(KWprint); return KWPRINT; }
 84 <INITIAL>"read"         { token(KWread); return KWREAD; }
 85 <INITIAL>"real"         { token(KWreal); return KWREAL; }
 86 <INITIAL>"string"       { token(KWstring); return KWSTRING; }
 87 <INITIAL>"then"         { token(KWthen); return KWTHEN; }
 88 <INITIAL>"to"           { token(KWto); return KWTO; }
 89 <INITIAL>"true"         { token(KWtrue); return KWTRUE; }
 90 <INITIAL>"return"       { token(KWreturn); return KWRETURN; }
 91 <INITIAL>"var"          { token(KWvar); return KWVAR; }
 92 <INITIAL>"while"            { token(KWwhile); return KWWHILE; }
 93
NORMAL  master  lex.l                                          31%   64:1
```

I have modified it to make lex scanner return what keyword and some other things required to be parsed.

2.What platfotm to run my parser

I use the Linux4 workstation to run my program, just SSH in it and run.

```
alfons@alfons:~|⇒  ssh afhwu0329@linux4.cs.nctu.edu.tw
afhwu0329@linux4.cs.nctu.edu.tw's password:
 _ _                    _ _          CPU: Intel(R) Xeon(R) CPU X5675 @ 3.07GHz
| (_)_ __  _   ___  _| || |         MEM: 7988 MiB  /tmp: 12K/4.0G
| | | '_ \| | | \ \/ / || |_
| | | | | | |_| |>  <|__   _|       Welcome to CS Linux Serivce!
|_|_|_| |_|\__,_/_/\_\  |_|         Open for all students and faculty

====[ Announcement of Computer Center, College of Computer Science, NCTU ]====
 1. Hostnames & IP Addresses of workstations :
    FreeBSD     : bsd1 ~ bsd6 (140.113.235.131 ~ 140.113.235.136)
                  alumni (140.113.235.116)
    Linux       : linux1 ~ linux6 (140.113.235.151 ~ 140.113.235.156)

 2. Useful Links:
       CCCS Duty Schedule <http://www.cs.nctu.edu.tw/schedule/>
       Frequently Asked Questions <http://www.cs.nctu.edu.tw/help/>

 3. For rights of other users, please don't occupy /tmp as yours,
    please use (re)nice/taskset/cpuset to lower the priority of high-loading processes,
    and please use ipcrm to clear shared memory after using it.
= Disk Usage ===================================================================
 Mail:                                                  0% 0/100000 KB

 Home:                                                  2% 6864/307200 KB

= Process ======================================================================
  PID TTY          TIME CMD
27380 pts/39   00:00:00 tcsh
27386 pts/39   00:00:00 csShell
27463 pts/39   00:00:00 ps
= Information ==================================================================
 Current Time: Tue Nov 21 15:15:51 CST 2017
 Online Users: 4
= CSCC Announce ================================================================

                                  CS Computer Center <help@cs.nctu.edu.tw>
```

And it successfully runs.

```
19:
20: end
21: end expr1

|------------------------------|
|  There is no syntactic error!  |
|------------------------------|
1: /**
2:  * statement.p: examples for each statement
3:  */
4: //&T-
5: statement;
6:
7: echo();
8: begin
9:         var s: string;
10:        read s;         // simple statement
11:        print s;        // simple statement
12: end
13: end echo
14:
15: funny(): boolean;
16: begin
17:        return 1<>1;    // return statement
18: end
19: end funny
20:
21:
22: begin
23:
24:        var a, b: real;
25:        var i, k: integer;
26:
27:        // simple statement
28:        a := 1.0;
29:        b := 2.3E-1;
30:
31:        // compound statement
32:        begin
33:                var c: boolean;
34:                c := a < b;
35:        end
```

3.How to run my parser

Makefile as the picture, description are written in the comments

```
TARGET = parser_exec #target_to_be output , which is the compiled file
OBJECT = lex.yy.c y.tab.h  y.tab.c #link the required libraries together
CC     = gcc -g #gcc -g for debugging oprions
CFLAGS = -Wall -Wextra -pedantic -g3 #compile flags
LEX    = flex #lex lib
LIBS   = -lfl #lfl flag
YACC = yacc -d -v #yacc flag

all: lex.yy.c y.tab.c #altogether
    $(CC) $(CCFLAGS) lex.yy.c y.tab.c -o $(TARGET) $(LIBS)

#generating the required components
y.tab.c: parser.y #compile parser
    $(YACC) parser.y

lex.yy.c: lex.l #compile lexer
    $(LEX) lex.l

.PHONY: clean

clean: #clean the file
    $(RM) -f $(TARGET) $(OBJECT)
```

4.Ability of my parser

Able to parse all the all testcases by TA and run through my own testcases
My own testcases--> https://pastebin.com/MZEvqiHW