

# Lab 2.1 -- Reinforce LEX and Use of User Defined Variables

Kay Sweebe  
CS370  
January 28, 2018

## Problem

A simple lex program was given that needed to be fixed. The goal of the program was to print the contents of a file except for what is written in comments. If any comments had errors, then the program is to print the debug error along with the line number that the error occurred on. Two things needed to be added. The first addition was that the format string %d needed to be passed an integer. The integer used was newline which is a counter that counts the current line number. The second addition was the directive for the newline character \n such that when a newline was found, the counter newline was incremented by one and a newline is printed.

## lab2remove.l

```
/* Kay Sweebe  
CS370  
Lab 2.1  
January 18, 2019
```

The changes I made were:

- To print the line number, the format string %d needed to be passed an integer. The integer used was newline which is a counter that counts the current line number.
- The directive for the newline character \n was added such that when a newline was found, the counter newline was incremented by one and we printed a newline.

```
*/
```

```
/* simple lex program which removes comments from a source program  
The main key is that a variable "comment" is set when the start of a comment
```

is seen and then unset when the ending set is seen. It is possible to have two starts closed by one end.

Shaun Cooper  
January 2015

```
*/
int comment = 0;
int debug=0; /* out prints debug statements if desired */
int newline = 1; /* counts number of newlines in input */
%%

"/*" {
    if (comment && debug) fprintf(stderr,
        " >>>>>>> line %d: Possible Nested comment <<<<<<<<\n", newline);
    comment = 1;
}

"*/" {
    if (!comment) printf("%s",yytext); /*print out if it is not in a comment */
    comment = 0;
}
. { if (!comment) printf("%s",yytext);}

\n {
    newline++; /* count number of lines in file */
    printf("\n");
}
%%

int yywrap(void)
{ return 1;
}

main()
{
    yylex();
}
```

# Makefile

```
# Kay Sweebe
# CS370
# Lab 2.1
#
# 1. Lex file
# Lex lab2remove.l outputs lex.yy.c
# 2. Compile file
# gcc -o remove lex.yy.c outputs remove

all:
# lex file
    lex lab2remove.l
# compile file
    gcc -o remove lex.yy.c
```

# Output

```
ksweebe@godel:~/Workspace/cs370/Lab2.1> ./remove < comment.tst

    and this represents lines of code.

    */ second termination

    see how it works
*/
```