

Compiler design lab

Mini PROJECT

SUBMMITTED BY:

**Abhishek M**

**(Reg. No:16030141CSE006)**

SYNOPSIS:

# **LEX program to add line numbers to a given file**

Lex is a computer program that generates lexical analyzers and was written by Mike Lesk and Eric Schmidt.  
Lex reads an input stream specifying the lexical analyzer and outputs source code implementing the lexer in the C programming language.

**Prerequisite:** [Flex (Fast lexical Analyzer Generator)](https://www.geeksforgeeks.org/flex-fast-lexical-analyzer-generator/)

A new line is encountered because of the **/n**. To count the number of lines, count the number of **/n** occurred with initial value as 1 as there exists an initial single line. All the other things can be ignored as focus is on number of **/n**. Take a counter initially set to 1 and increment it whenever a new line (/n) occurs.

* s it a capital letter in English? [A-Z] : increment capital letter count by 1.
* Is it a small letter in English? [a-z] : increment small letter count by 1
* Is it [0-9]? increment digit count by 1.
* All other characters (like '!', ['@','&'](mailto:'@','%26')) are counted as special characters
* How to count the number of lines? we simply count the encounters of '\n' <newline> character.that's all!!
* To count the number of words we count white spaces and tab character(of course, newline characters too..)

SOFTWARE USED:

Ubuntu

Ubuntu is an open-source operating system (OS) based on the Debian GNU/Linux distribution.

Ubuntu incorporates all the features of a Unix OS with an added customizable GUI, which makes it popular in universities and research organizations. Ubuntu is primarily designed to be used on personal computers, although a server editions does also exist.

Ubuntu was first released in 2004. The project is sponored by Canonical Ltd., a U.K.-based company that generates revenue by selling support and services to complement Ubuntu. Canonical releases a new version of Ubuntu every six months and provides support in the form of patches and security releases for 18 months thereafter.

Ubuntu consists of many software packages, which are licensed under GNU General Public License. This allows users to copy, change, develop and redistribute their own version of the program.

Ubuntu comes with a wide range of software programs, including FireFox and LibreOffice. There is also proprietary software that can be run on Ubuntu.

# **LEX program to extract HTML tags from a HTML file:**

AIM:

To write a program to extract HTML tags from HTML file

ALGORITHM:

Step1: Lex program contains three sections: definitions, rules, and user subroutines. Each section must be separated from the others by a line containing only the delimiter, %%. The format is as follows: definitions %% rules %% user\_subroutines

Step2: In definition section, the variables make up the left column, and their definitions make up the right column. Any C statements should be enclosed in %{..}%. Identifier is defined such that the first letter of an identifier is alphabet and remaining letters are alphanumeric.

Step3: In rules section, the left column contains the pattern to be recognized in an input file to yylex(). The right column contains the C program fragment executed when that pattern is recognized. The various patterns are keywords, operators, new line character, number, string, identifier, beginning and end of block, comment statements, preprocessor directive statements etc.

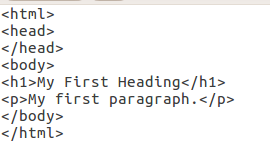
Step4: Each pattern may have a corresponding action, that is, a fragment of C source code to execute when the pattern is matched.

Step5: When yylex() matches a string in the input stream, it copies the matched text to an external character array, yytext, before it executes any actions in the rules section.

Step6: In user subroutine section, main routine calls yylex(). yywrap() is used to get more input.

Step7: The lex command uses the rules and actions contained in file to generate a program, lex.yy.c, which can be compiled with the cc command. That program can then receive input, break the input into the logical pieces defined by the rules in file, and run program fragments contained in the actions in file.

Input File:



PROGRAM CODE:

/\* Declaration section\*/

%{

%}

%%

"<"[^>]\*> {printf("%s\n", yytext); }  /\* if anything enclosed in

                                      these < > occur print text\*/

. ;  // else do nothing

%%

int yywrap(){}

int main(int argc, char\*argv[])

{

    // Open tags.txt in read mode

    extern FILE \*yyin = fopen("tags.txt","r");

    // The function that starts the analysis

    yylex();

    return 0;

}

RESULT:

Thus the program for implementation of LexicalAnalyzer using Lex tool has been executed successfully.

OUTPUT:

