Pritam Mogal TE (B) -17

Assignment No. 6

LEX Program

Aim: Design LEX program to count no. of words, lines and characters of given input file.

Problem Statement: Write a program using LEX specifications to implement lexical analysis phase of compiler to count no. of words, lines and characters of given Input file.

Pre-requisites :- LEX Basics.

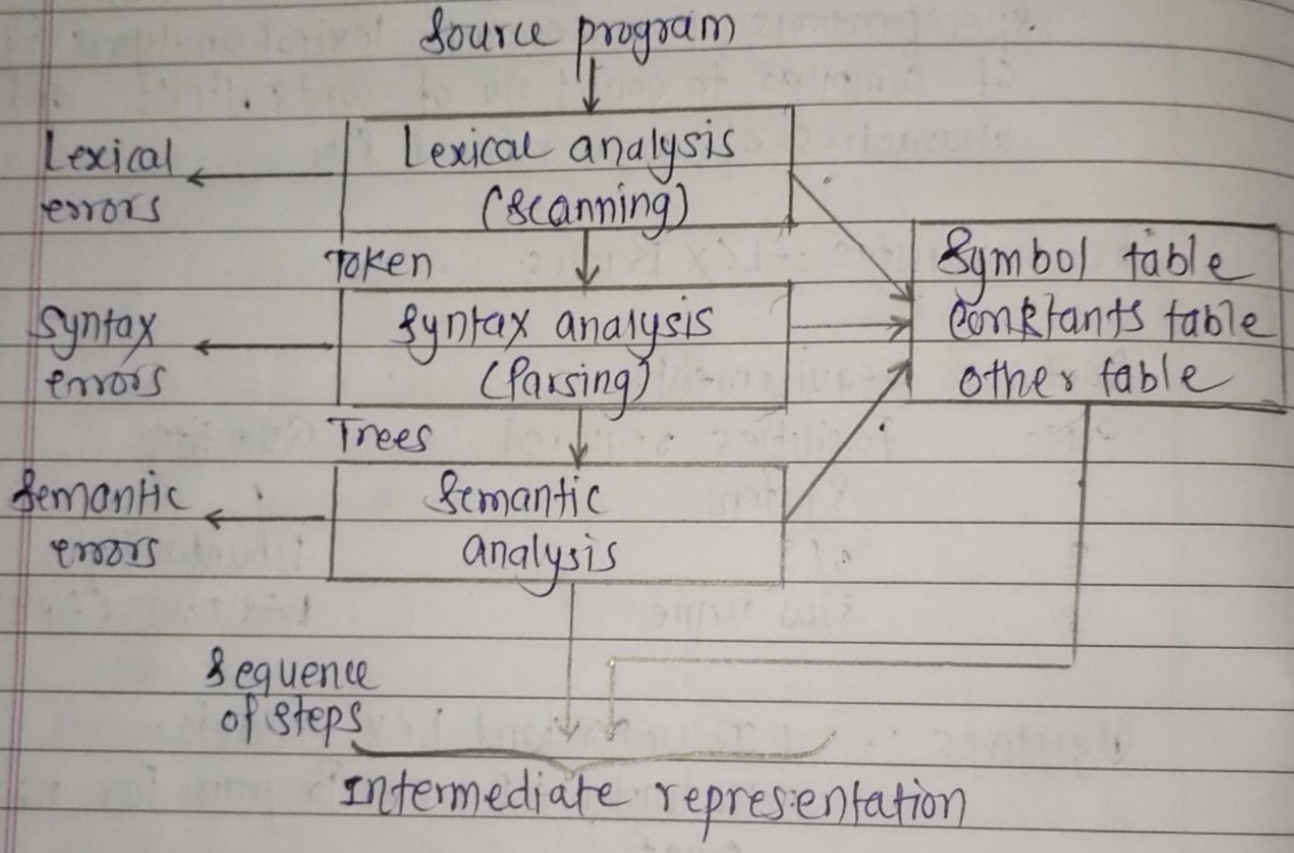
Software requirements:

S.No	Facilities required	Quantity
1	System	1
2	O/S	Ubantu Kylin
3	S/w name	LEX Tool (flex)

Objectives :- 1. To understand LEX Concepts.
2. To implement LEX Program for no's of count
3. To study about Lex & Java.
4. To know important about lexical analyzer.

How the input is matched: When the generated Scanner is run, it analyzes its input looking for strings, which match any of its patterns. It is finds more than one match, it takes the one matching the most text. It is finds two or more matches of the same length, the rule listed first in the flex input file is choosen. Once the match is determined,

The text corresponding to the match (called the token) is made available in the global character pointer "yytext", and its length in the global integer "yyleng".



Front end of toy compiler.

Conclusion:

Thus, we have studied lexical analyzer and implemented an application for lexical analyzer to count total number of words, chars and line etc.

Name: Pritam Gotiram Mogal **DivRoll**: TE –B-17

Assignment No. 06 [LEX Program]

Problem Satement: Write a program using Lex specifications to implement lexical analysis Phase of compiler to count no. of words, lines and characters of given Input file.

---

1. Code b3.l:

```
%{
int no_line=0; int
no_space=0; int
no_char=0; int
no_words=0;
#include<string.h>
%}

%%
([ a-zA-Z])+ {no_words++; no_char+=strlen(yytext); }
[ " "] {no_space++; }
[ "\n"] {no_line++; }
. ;

%%


int yywrap(){

}

int main(int argc,char* argv[]){
yyin=fopen("test.txt","r");
   yylex();
   printf("Total Spaces %d\n",no_space);
printf("Total Words %d\n",no_words);
printf("Total Line %d\n",no_line);
no_char+=no_space;
   printf("Total Char %d\n",no_char);
   fclose(yyin);
}
```

2. text.txt File:

// Content of text.txt File

The earliest foundations of what would become computer science predate the invention of the modern digital computer. Machines for calculating fixed numerical tasks such as the abacus have existed since antiquity, aiding in computations such as multiplication and division. Algorithms for

performing computations have existed since antiquity, even before the development of sophisticated computing equipment.
Computer science, the study of computers and computing, including their theoretical and algorithmic foundations, hardware and software, and their uses for processing information. The discipline of computer science includes the study of algorithms and data structures, computer and network design, modeling data and information processes, and artificial intelligence. Computer science draws some of its foundations from mathematics and engineering and therefore incorporates techniques from areas such as queueing theory, probability and statistics, and electronic circuit design. Computer science also makes heavy use of hypothesis testing and experimentation during the conceptualization, design, measurement, and refinement of new algorithms, information structures, and computer architectures.

OUTPUT:

```
 Pritam-spos@Pritam-HP:~/SPOSL/LexProgram$ lex b3.l

Pritam-spos@Pritam-HP:~/SPOSL/LexProgram$ gcc lex.yy.c

Pritam-spos@Pritam-HP:~/SPOSL/LexProgram$  ./a.out test.txt

Total Spaces 155

Total Words 157

Total Line 3

Total Char 1180
```