

Assignment - 08

* Problem Statement :- Write a program to implement a lexical analyzer for a subset of 'C' language.

* Objective :-
1. To understand the basic principles in compilation
2. To study lexical analysis phase of compiler.

* Theory :-

Compiler takes ip as a source program and produces output as an equivalent sequence of machine instructions. This process consists of 2 step processing of source program →
1. Analysis
2. Synthesis

The Analysis phase consists of 3 sub-steps :

- i) lexical
- ii) syntax
- iii) semantic

i) lexical analysis : The action of scanning the source program into proper syntactic classes is known as lexical analysis.

• Task of lexical analysis →

1. To scan the program into basic elements or tokens of the language.
2. To build the uniform symbol table
3. To build the symbol and literal table.
4. To remove white spaces and comments



33304

myCOMPANION

5. To detect errors such as invalid identifier on constant.

• Data Structures \rightarrow

1. Source program -

Original source program which is scanned by compiler as a string of characters.

2. Terminal table -

It stores arithmetic operators, keywords etc

3. Literal table -

It stores all literal and addresses

4. Identifier table

5. Uniform Symbol Table

• Algorithm \rightarrow

1. Initialize line no. to 1

2. Read the source program line by line (in buffer)

3. For each line separate the tokens such as

(i) Identifier / function names / keywords :-

Follow the transition diagrams to detect this i.e. letter followed by letter or digit. Search in keyword table for existence of keyword, otherwise it is identifier or function name

(ii) Integer constant : digit followed by digit

(iii) All types of operators such as $>$, $<$, $++$, $+$ etc

(iv) Remove comments of type $/* \dots */$.

33304



DATE / /
PAGE NO

myCOMPANION

① Remove all white spaces.

4. Assign a line no. and increment line no.

5. Repeat steps ② to ④ till end of file.

* Conclusion :- Thus in this ~~this~~ assignment, we learnt ~~about~~ about the lexical analysis phase of a compiler and implemented a program to replicate the working of a lexical analyzer.