

ASSIGNMENT - 7

TITLE : LEXICAL ANALYSER

PROBLEM STATEMENT: Write a program to implement a lexical analyser for a subset of 'C' language.

THEORY:

Compiler takes input as source program & produces output as an equivalent sequence of machine instructions.

This step consists of two steps: processing of source program

1. Analysis of source program:

It consists of 3 sub-steps:

i. Lexical analysis - determine lexical constituents in source program

ii. Syntax phase - determine structure of source string

iii. Semantic analysis - determine meaning of source string.

2. Synthesis of target program:

It deals with memory allocation & code generation. The actions in analysis phase are uniquely defined for a given language. Synthesis step contains of many action-instances that depend on the aspect concerning the execution environment of the computer.

Lexical analysis:

The action of scanning the source program into the proper syntax classes is known as lexical analysis.

Task of lexical analyser:

- i. To scan the program into basic elements or tokens of the language
- ii. To build the uniform symbol table.
- iii. To build symbol & literal table
- iv. To remove whitespace & comments
- v. To detect errors such as invalid identifier & constant

Data Structures:

- i. Source program - original source code scanned by compiler as string of characters
- ii. Terminal table - a permanent database that has entry for each terminal symbol.
- iii. Literal table - created to describe all literals in the program.
- iv. Identifier table - Describes all identifiers in the table
- v. Uniform Symbol Table - created during lexical analysis to repeat the program as a string of tokens, rather than of individual characters.

Each VST contains identification of the table to which it belongs

```
#include <iostream>
using namespace std;
main()
{
    int i = 100;
    printf("Hello World");
    printf("%d", i);
}
```

Terminal Table	Identifier Table	Literal Table
1 (1 main	1 100
2)	2 printf	2 Hello World
3 {	3	3 %d
4 =		
5 ;		
6 }		
7 /		
8 int		
9 }		

VST

1	main	IDN	1	7	=	TRM	4
2	(TRM	1	8	100	LIT	1
3)	TRM	2	9	;	TRM	5
4	{	TRM	3	10	printf	IDN	3
5	int	TRM	8	11	"Hello World"	LIT	2
6	i	TRM	81	12)	TRM	2

13)	TRM	2
14	;	TRM	5
15	printf	IDN	3
16	(TRM	1
17	"%d"	LIT	3
18	,	TRM	7
19	i	IDN	2
20)	TRM	2
21	;	TRM	5
22	}	TRM	6

Conclusion:

In this assignment, we have implemented lexical analyzer for a subset of 'C' language.