Manil S. Malla

wcrecent@gmail.com

Constructing Symbol table for reserved keywords and other language constructs

Compiler Lab

Constructing Symbol table

* **Introduction**
* Symbol table [From notes]
* **Objective**
* WAP to construct symbol table that handles reserved keywords and other language constraints.

|  |  |  |  |
| --- | --- | --- | --- |
| **lexptr** | **token** | **attributes** |  |
|  |  |  | 0 |
| i | If |  | 1 |
| e | Else |  | 2 |
| c | id |  | 3 |
| i | id |  | 4 |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| i | f | EOS | e | l | s | e | EOS | c | o | u | n | t | EOS | i | EOS | … | … | … | … |

Lexemes

Fig: - Symbol table and array for storing strings

* Program design

Init.c [Notes pg 89]

Symbol.c [Notes pg 88]

* Operations associated with symbol table
* Insert(s,t) - returns index of new entry foe lexeme ‘s’ and tokens ‘t’.
* Lookup(s) - returns index of the entry for string ‘s’ , or 0 if s isn’t found.
* Program design

struct entry

{

Char \*lexptr;

Int token;

}

Char lexemes [STRMAX];

Struct entry symtable [STRMAX];

Int lastchar = -1;

Int lastentry = 0;

* Program constants
* #define EOS ‘\0’