

GROUP -5

CD LAB ASSIGNMENT-3

ANALYSIS

Implementation of Symbol Table:

Symbol Table is an important data structure created and maintained by the compiler in order to keep track of semantics of variable i.e. it stores information about scope and binding information about names, information about instances of various entities such as variable and function names, classes, objects, etc.

Using Linear List:

- >> In this method, an array is used to store names and associated information.
- >> A pointer “available” is maintained at end of all stored records and new names are added in the order as they arrive
- >> To search for a name we start from beginning of list till available pointer and if not found we get an error “use of undeclared name”
- >> While inserting a new name we must ensure that it is not already present otherwise error occurs i.e. “Multiple defined name”
- >> Insertion is fast $O(1)$, but lookup is slow for large tables – $O(n)$ on average
- >> Advantage is that it takes minimum amount of space.

Using Linked List:

- >> This implementation is using linked list. A link field is added to each record.
- >> Searching of names is done in order pointed by link of link field.
- >> A pointer “First” is maintained to point to first record of symbol table.
- >> Insertion is fast $O(1)$, but lookup is slow for large tables – $O(n)$ on average

Using Hash Table:

>>In hashing scheme two tables are maintained – a hash table and symbol table and is the most commonly used method to implement symbol tables..

>>A hash table is an array with index range: 0 to $\text{tablesize} - 1$. These entries are pointer pointing to names of symbol table.

>>To search for a name we use hash function that will result in any integer between 0 to $\text{tablesize} - 1$.

>>Insertion and lookup can be made very fast – $O(1)$.

>>Advantage is quick search is possible and disadvantage is that hashing is complicated to implement.

=====