SRM UNIVERSITY AP
Andhra Pradesh

Compiler Design Lab
CSE306L

Group-6

Lab Report

Implementation of Symbol Table using Linked List

Symbol Table:

The compiler creates and maintains a data structure to store information about the occurrence of various entities such as variable and function names, objects and classes, etc. This kind of data structure is known as a "Symbol table".

Items stored in the symbol table:

- Variable names and constants
- Procedure and function names
- Literal constants and strings
- Compiler generated temporaries
- Labels in source languages

The symbol table is an important data structure created and maintained by compilers in order to store information about the occurrence of various identifiers such as variable names, function names, objects, classes, interfaces, etc. The symbol table is used by a compiler's analysis and synthesis parts. The symbol table can be implemented in one of the following ways:

- Linear (sorted or unsorted) list
- Binary Search Tree
- Hash table
- And other ways.

Implementation of Symbol Table using Linked List:

Insertion time:

In order to search for the position to insert the element traversing is required. Thus, it takes O(n) time.

Lookup Time:

While fetching a data item the linked list must be traversed completely (linear search). Thus, this operation takes O(n) time.

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

The main use of a linked list is to realize the following functions of the symbol table:

- size() - Get the size of the symbol table.
- show_items() - Displays the elements in the symbol table in dictionary form.
- put() - Puts the element into the symbol table, you need to consider the order of the key, and the value of the repeated key will be replaced.
- delete() - Delete the specified node according to the key.
- get_item() - Gets the value of the corresponding node through the key.

Advantages and Disadvantages of Linked Lists:

A few benefits of using linked lists are:

- Nodes can be easily deleted and inserted.
- A linked list can be easily implemented in most programming languages.

On the contrary, some limitations of linked lists are:

- Nodes must always be accessed sequentially, which is time-consuming.
- The pointers used in linked lists require additional memory.

Disadvantages of using Array, Linked Lists, or Unsorted Lists for symbol table implementations :

- Lookup time is directly proportional to the table size.
- Every insertion operation is preceded by a lookup operation.