21163

Page No.______
Date: / /

Assignment 7

Title: Symbol Table Hashing

Problem Statement:
It is generated by compiler. From this perspective a symbol table is a set of name-attribute pairs. Perform the following operations on symbol table:
1) Determine if particular name is in symbol table.
2) Retrieve the attributes of that name.
3) Modify the attributes of that name.
4) Insert a new name and it's attributes.
5) Delete a name & it's attributes.

Objective:
To understand the concept of chaining used in Hash tables.

Outcome:
It will be able to implement symbol table structure using hashing & chaining with & without replacement.

Requirements:
1) 64 bit machine, operating system.
2) Editor, Compiler (g++)
3) CPU, RAM

Theory:

In linear probing technique if a large number of keys are being mapped to the same location then searching time complexity no longer remains O(1) but it can take O(n) time for search. Chaining technique tries to reduce time. In chaining all keys which get the same location on applying hash function are stored in chain.

- Pseudo Code:

```
• add (symbol a)
  if (replac)
          if hash ( s[b].k != b) and s[b] non Empty
              swap (s[b], a)
          ek if s[b] non empty
              for c = b  until c = -1
                  c = s[c].chain
              l :-- First empty loc after b
                  if (no loc.)
                      error
                  else
                      s[l] := a
                      s[c].chain := l
```

21163

Page No.______
Date: / /

```
else  from l:=b
          while (s[l] not empty & hash s[x].k != b){
                if (all loc. non empty)
                      print'error' & return
          }
          if s [l] = empty
               s[l] = a    ret
          if c:=l  until   s[c]. chain = -1
               c:= s[c]. chain
           m := next  empty  after c
               if (not found)
                 print 'error', return
          else
               s[m]:= a
               s [c].chain := m

retrieve (symbol a)
     b:= find (a.k)
     if (b= -1)
          print "not found"
     else
          Read attr. of s[b]
          print attr.
return.
```

Scanned with CamScanner

21163

Page No.______
Date: / /

- update (symbol a)

```
b:= find (a.key)
if b = -1
        print "Not found" & return
else
        read attr of  s[b] from user
return.
```

- delete (symbol a)

```
b:= find (a.key)
if (b = -1)
        print (not found) ret
else
        l = loc. s.t (s[l].chain = b)
        if (loc = found)
                s[l].chain = S[b].chain
        make s[b] = empty
return.
```

* Problems. faced

1. Probing without replacement
2. Collision resolution.

Page No.
Date: / /

- Teot cases:

|  | Input | Operation | Output | Reoult |
|---|---|---|---|---|
| 1) | Abc, Ade, Bcd, Bde, Abcd, Def (without Replacement) | find (Abc) | loc: 0 | Success |
|  |  | find (Bcd) | loc: 2 |  |
|  |  | find (Def) | loc: 5 |  |
|  |  | find (Abcd) | loc: 4 |  |
|  |  | delefe (Ade) | deleted |  |

- Conclusion:
Symbol table is implemented using hashing using chaining with & without replacement.