



Hashing

Öğrenci Adı: İclal Ertürk

Öğrenci Numarası: 21011037

Dersin Öğretmeni: M. Amaç GÜVENSAN

Video Linki: <https://youtu.be/3Jf9F3epvi0>

1- Problemin Çözümü:

Kullanıcıdan ilk olarak sembol tablosunu oluşturmak için bir dosya adı ve çalışma modu alınmıştır. Kullanıcı tarafından verilen dosyada değişkenlerin sayısı tespit edilmiştir. Tespit edilen değişken sayısının iki katından büyük olan en küçük asal sayı, tablo boyutu olarak belirlenmiştir. Tablo için yer ayrılmış ve elemanlara null atanmıştır.

Değişken adının karakter değerleri kullanılarak Horner methodu ile hash değeri hesaplanmıştır. Double hashing için 2 hash fonksiyonu yazılmış ve bunların birleştiği 3. Hash fonksiyonu yazılmıştır. Hashin sonucu değişkenin tabloda yerleşeceği yeri vermektedir.

Dosyadan okunan veri “ ; \n ” e göre tokenlere ayrılmıştır ve token char, int veya floats a bir değişken ataması gerçekleştirilmiştir. Bu değişken tiplerinden sonradaki değişkenleri bulmak için _ ile başlayan tokenlerin varlığına bakılmıştır. Bu değişken sembol tablosunda yok ise tabloya eklenmiştir. Eklendiği adres ve hesaplanan ilk adres struct içerisinde tutulmuştur. Eğer değişken tabloda bulunuyorsa eklenmemiştir. Bu işlem insert fonksiyonu içinde gerçekleştirilmiştir. Insert içerisinde lookup fonksiyonu ile değişkenin sembol tablosunda olup olmadığı kontrol edilmiştir. Tokenlerin değişken tipi olmadığı durumda “_” karakteri ile başlayıp başlamadığı kontrol edilmiştir. “_” ile başlayan tokenler bir değişkendir ve deklere ediliyor edilmedikleri kontrol edilmiştir.

Debug modunda, sembol tablosunun son hali, tablo boyutu, deklere edilmiş değişkenlerin sayısı tabloya yerleşen değişkenlerin yerleşmesi için hesaplanan ilk hash değeri ile yerleştiği adres ve hatalar kullanıcıya gösterilmiştir. Normal modda sadece hatalar kullanıcıya gösterilmiştir.

2- Karşılaşılan Sorunlar:

Tablo boyutu hesaplarken ve tabloya değişkenlerin eklenip eklenmeyeceğinin kontrolünü yaparken benzer kod blokları kullanılmıştır. Bunu önlemek için calculateTableSize adında bir fonksiyon üretilmiştir fakat aynı kod kısmının tekrarlanması önüne tam olarak geçilememiştir.

Horner fonksiyonu ile sonuç üretirken değişkenin decimal karşılığı büyük olduğu için değişkenin harf sayısı büyük olduğu zaman overflow meydana geliyor ve negatif bir sonuç üretiliyor. Bunun için her harften ‘A’ çıkartılmıştır. Bu durumda da değişken çok fazla sayı içermediği sürece negatif bir sonuçla karşılaşmamıştır.

Debug modunda istenilenlerden ilk madde olan deklere edilmiş değişken sayısı isteniyor. Burada tekrar eden değişkenin hesaplanıp hesaplanmayacağı tam anlaşılamamıştır. Bu nedenden ötürü tekrar etmeyenlerin de tüm deklere edilen değişkenlerin de sayısı yazdırılmıştır.

Bunlar haricinde uçlarda bir sorunla karşılaşmamıştır.

3- Karmaşıklık Analizi:

Type: okunan değişkenin türü

Yerlesenler: tabloya yerleşen değişkenin adını, hesaplanan ilk hash değerini ve yerleştiği indisi tutan struct

WHILE dosyadan satır okunabiliyorsa oku:

```
CALL Strtok
```

```
IF token != NULL THEN
```

```
IF token = "char" or "int" or "float" THEN
```

```
    Type = token
```

```
    Token=strtok(NULL, " ,;\n")
```

```
    WHILE token != NULL:
```

```
        IF token[0]='_' THEN
```

```
            If CALL insert=0 THEN
```

```
                Print değişken daha önce deklere edilmiş
```

```
            ELSE:
```

```
                Yerlesenler.name=name
```

```
                Yerlesenler.first=first
```

```
                Yerlesenler.last=last
```

```
                Token=strtok(NULL, " ,;\n")
```

```
            ENDIF
```

```
        ENDIF
```

```
    END WHILE
```

```
ELSE:
```

```
    WHILE token !=NULL:
```

```
        IF token[0]='_' THEN
```

```
            IF CALL lookup =0
```

```
                Printf token deklere edilmemis
```

```
            ENDIF
```

```
        ENDIF
```

```
    END IF
```

```
ENDIF
```

END WHILE

Algoritma da bulunan en karmaşık bölge dosyadaki kelimeleri okuyup tabloya yerleştirilip yerleştiremeyeceğinin kontrolünün yapıp gerçekleştirildiği bölgedir. Yukarıda pseudo kodu verilen algoritma bu bölgeyi göstermektedir.

Insert fonksiyonu içinde lookup fonksiyonu çağırılmıştır ve ek karmaşıklık getirecek bir durum yoktur. bu durumda $O(\text{insert}) \in O(\text{lookup})$

Aşağıda lookup ın sözde kodu verilmiştir.

Lookup:

Input: name: aranacak değişkenin adı

Table: sembol tablosu

tableSize: tablonun boyutu

hashResult: değişkenin en son hesaplanan hash sonucu

firstHash: değişkenin ilk hesaplanan hash sonucu

i=0

key = horner with name

CALL sprintf with sayi, key

hashResult = hash(key, i, tableSize)

firstHash = hashResult

WHILE i < tableSize AND table[hashResult][1] != "null":

 IF table[hashResult][1] = sayi: THEN

 i=tableSize

 RETURN 1 // Variable found

 ELSE:

 i++

 hashResult = hash(key, i, tableSize)

 END IF

END WHILE

IF table[hashResult][1] = "null":

 RETURN 0

ELSE:

 RETURN 1

END IF

Hash fonksiyonu değişkenden bağımsız hesap yaptığı için karmaşıklığı $O(1)$ 'dir. Horner fonksiyonunun karmaşıklığı aranan değişkenin uzunluğuna bağlı olarak değişir. Değişkenin uzunluğu N olarak varsayıldığında Horner fonksiyonunun karmaşıklığı:

$$\sum_{0}^{N-1} 1 = N*(N-1)/2 \cong N \cong \theta(N)$$

Lookup fonksiyonunun geri kalanına bakıldığında karmaşıklık en kötü durumda tableSize'a eşit olmaktadır. En iyi ve ortalama durumda ise $O(1)$ e eşit olmaktadır. Çoğu zaman elemanlar ilk seferde bulunmaktadır. Ekstra bir işlem yapılmadığından kalan kısmı karmaşıklığına $O(1)$ denilebilmektedir.

Sonuç olarak lookup'ın karmaşıklığı:

$$\theta(N) + \theta(1) \cong \theta(N)$$

Bu durumda insert fonksiyonunun karmaşıklığı da $\cong \theta(N)$ 'dir. Yazılan en karmaşık kod bloğunun karmaşıklığı ise :

$\text{insert} * \text{tokenSayisi} \cong \theta(N * \text{tokenSayisi})$ olarak ifade edilebilir.

4- Ekran Çıktıları:

1.1-

```
1.NORMAL MOD
2.DEBUG MOD
CHOOSE MODE: 1
Enter the file name: ornek.c

_aa daha once deklere edilmiş
_xx deklere edilmemiş
_dd deklere edilmemiş
-----
Process exited after 5.19 seconds with return value 0
Press any key to continue . . . |
```

1.2-

```
1.NORMAL MOD
2.DEBUG MOD
CHOOSE MODE: 2
Enter the file name: ornek.c

_aa daha once deklere edilmis
_xx deklere edilmemis
_dd deklere edilmemis

Deklere edilmis tekrar etmeyen degisken sayisi: 4
Deklere edilmis tum degiskenlerin sayisi: 5
Sembol tablosunun uzunlugu: 11

Deklere edilen her degiskenin hesaplanan ilk adresi ve yerlestigi adres:
-----
| name | ilk | son |
-----
| _aa | 0 | 0 |
-----
| _bb | 10 | 10 |
-----
| _cc | 9 | 9 |
-----
| _x | 6 | 6 |
-----

Sembol tablosunun son hali:
-----
| satir | type | variable |
-----
| 0 | int | 29854 |
-----
| 1 | null | null |
-----
| 2 | null | null |
-----
| 3 | null | null |
-----
| 4 | null | null |
-----
| 5 | null | null |
-----
| 6 | char | 985 |
-----
| 7 | null | null |
-----
| 8 | null | null |
-----
| 9 | int | 29918 |
-----
| 10 | int | 29886 |
-----

-----
Process exited after 6.668 seconds with return value 0
Press any key to continue . . .
```

2.1-

```
C:\Users\icla\OneDrive\Masa  X + v
1.NORMAL MOD
2.DEBUG MOD
CHOOSE MODE: 1
Enter the file name: ornek2.c

_iclal daha once deklere edilmis
_aa deklere edilmemis
_xx deklere edilmemis
_bb deklere edilmemis
-----
Process exited after 6.097 seconds with return value 0
Press any key to continue . . . |
```

2.2-

```
C:\Users\icla\OneDrive\Masa  X + v
1.NORMAL MOD
2.DEBUG MOD
CHOOSE MODE: 2
Enter the file name: ornek2.c

_iclal daha once deklere edilmis
_aa deklere edilmemis
_xx deklere edilmemis
_bb deklere edilmemis

Deklere edilmiş tekrar etmeyen degisken sayisi: 4
Deklere edilmiş tüm degiskenlerin sayisi: 5
Sembol tablosunun uzunlugu: 11

Deklere edilen her degiskenin hesaplanan ilk adresi ve yerlestigi adres:
-----
| name | ilk | son |
-----
| _iclal | 10 | 10 |
-----
| _ayse | 10 | 0 |
-----
| _asma | 5 | 5 |
-----
| _masa | 1 | 1 |
-----

Sembol tablosunun son hali:
-----
| satir | type | variable |
-----
| 0 | int | 28714344 |
-----
| 1 | char | 29048768 |
-----
| 2 | null | null |
-----
| 3 | null | null |
-----
| 4 | null | null |
-----
| 5 | char | 28708388 |
-----
| 6 | null | null |
-----
| 7 | null | null |
-----
| 8 | null | null |
-----
| 9 | null | null |
-----
| 10 | int | 896870622 |
-----

Process exited after 4.986 seconds with return value 0
Press any key to continue . . . |
```

3.1-

```
C:\Users\icla\OneDrive\Masa  X + v
1.NORMAL MOD
2.DEBUG MOD
CHOOSE MODE: 1
Enter the file name: ornek3.c

_aa daha once deklere edilmis
_k daha once deklere edilmis
_abc deklere edilmemis
_bb deklere edilmemis
_dd deklere edilmemis
_xx deklere edilmemis
-----
Process exited after 8.956 seconds with return value 0
Press any key to continue . . . |
```

3.2-

```
1.NORMAL MOD
2.DEBUG MOD
CHOOSE MODE: 2
Enter the file name: ornek3.c

_aa daha once deklere edilmis
_k daha once deklere edilmis
_abc deklere edilmemis
_bb deklere edilmemis
_dd deklere edilmemis
_xx deklere edilmemis

Deklere edilmis tekrar etmeyen degisken sayisi: 6
Deklere edilmis tum degiskenlerin sayisi: 8
Sembol tablosunun uzunlugu: 17

Deklere edilen her degiskenin hesaplanan ilk adresi ve yerlestigi adres:

| name | ilk | son |
|_aa | 2 | 2 |
|_cc | 15 | 15 |
|_x | 16 | 16 |
|_y | 0 | 0 |
|_z | 1 | 1 |
|_k | 3 | 3 |

Sembol tablosunun son hali:

| satir | type | variable |
| 0 | int | 986 |
| 1 | int | 987 |
| 2 | char | 29854 |
| 3 | int | 972 |
| 4 | null | null |
| 5 | null | null |
| 6 | null | null |
| 7 | null | null |
| 8 | null | null |
| 9 | null | null |
| 10 | null | null |
| 11 | null | null |
| 12 | null | null |
| 13 | null | null |
| 14 | null | null |
| 15 | char | 29918 |
| 16 | int | 985 |

-----
Process exited after 7.614 seconds with return value 0
Press any key to continue . . . |
```


4.1-

```
C:\Users\icla\OneDrive\Masa  X + v
1.NORMAL MOD
2.DEBUG MOD
CHOOSE MODE: 1
Enter the file name: ornek4.c

_kalem daha once deklere edilmis
_xx deklere edilmemis
_a deklere edilmemis
-----
Process exited after 5.812 seconds with return value 0
Press any key to continue . . . |
```

4.2-

```
C:\Users\icla\OneDrive\Masa  X + v
1.NORMAL MOD
2.DEBUG MOD
CHOOSE MODE: 2
Enter the file name: ornek4.c

_kalem daha once deklere edilmis
_xx deklere edilmemis
_a deklere edilmemis

Deklere edilmiş tekrar etmeyen değişken sayısı: 5
Deklere edilmiş tüm değişkenlerin sayısı: 6
Sembol tablosunun uzunluğu: 13

Deklere edilen her değişkenin hesaplanan ilk adresi ve yerleştiği adres:
-----
| name | ilk | son |
-----
| _kalem | 5 | 5 |
-----
| _kelam | 4 | 4 |
-----
| _aa | 6 | 6 |
-----
| _bb | 12 | 12 |
-----
| _alem | 12 | 11 |
-----

Sembol tablosunun son hali:
-----
| satir | type | variable |
-----
| 0 | null | null |
-----
| 1 | null | null |
-----
| 2 | null | null |
-----
| 3 | null | null |
-----
| 4 | int | 898777247 |
-----
| 5 | int | 898658207 |
-----
| 6 | char | 29854 |
-----
| 7 | null | null |
-----
| 8 | null | null |
-----
| 9 | null | null |
-----
| 10 | null | null |
-----
| 11 | char | 28701425 |
-----
| 12 | char | 29886 |
-----

Process exited after 5.535 seconds with return value 0
Press any key to continue . . . |
```