

PROGRAMLAMA LABORATUVARI 1

1. PROJE

Mohamed Hosam Mohamed Gomaa Helwa

Bilgisayar Mühendisliği Bölümü
Kocaeli Üniversitesi

190201140@kocaeli.edu.tr

ÖZET

Bu doküman Programlama Laboratuvarı Dersi 1. Bu proje C dili kullanarak oluşturdum. Binary Search algoritmasıyla index dosyasına insert veya search uygulandı. Dökümanda projenin tanımı, çözüme yapılan algortmaları, kullanılan yöntemler, akış şemeleri ve kullandığım fonksiyonların özeti açıklanılmıştı.

Dizin dosyasında İkili Arama.



- İndeks dosyasındaki sayıları karşılaştırmak için önce indeks dosyasında onlara ulaşmam gerekiyordu.
- afterNewlineNumber(int*ptr)Fonksiyon kullanarak, bir pointer index dosyasında herhangi bir yere veriyorum ve fseek() ile "n"yi bulana kadar pointer pointer geri dönecek, sonra bulunan sayıyı return olarak veriyor.
- Binary Search döngüsünü kırmak için, ekleme alanını bulduğumda, ilk ve son işaretçinin aynı sayıyı göstererek tekrar ettiğini öğrendim, bu yüzden tekrar ettiğinde döngüyü kırıyorum.

Veri dosyasını indeks dosyasıyla bağlamak

- Verileri binary dosyasına kaydetmeden önce fseek() kullanarak dosyanın son pointer ne olduğunu bir değişkene kaydediyorum. Deyilem ki bu değişkenin adı "index".
- Son pointer öğrendikten sonra verileri binary dosyasında saklarım.
- Öğrenci numarasını ve index numarasını eklemek için index dosyasına gönderiyorum.
- searchIndexes() Fonksiyonu kullanarak, ekleme noktasının işaretçiyi getiriyorum.
- seachIndexes() Fonksiyon dönüş değerine bağlı olarak, bunları doğru konuma eklemek için insertByPtr() Fonksiyonu kullandım.
- Kayıt nasıl eklediğini "kayıt eklemek" kısmına daha çok detaylar var.

"searchIndexes()"Insert olarak

- Bu Fonksiyon, index dosyasına öğrenci numarası ve dizin eklemek ve ayrıca index dosyasında bir sayı aramak için de kullanıldı.

- İndeks dosyasındaki ekleme noktasını hızlı bir şekilde bulabilmek için ikili aramayı kullanılmıştı.

- Bu fonksiyon 4 bölüme ayrıldı :

- Eğer index dosyası boş ise, öğrenci numarasını ve index doğrudan dosyaya eklemem gerektiği anlamına gelir.
- Eğer dosyada sadece bir satır varsa, mevcut öğrenci numarası ve eklemek istediğim öğrenci numarası ile karşılaştırmam gerekmektedir, böylece cevaba bağlı olarak o satırdan önce veya sonra ekleyeceğim.
- Eğer öğrenci numarası:
 - İlk öğrenci numarasından küçükse veya eşitse, bu ondan önce ekleme anlamına gelir.
 - Son öğrenci numarasından büyükse veya eşitse, bu ondan sonra ekleme anlamına gelir.
 - İlk öğrenci numarasından büyükse veya eşitse ve ikinci öğrenci numarasından küçükse, ilk satırdan sonra ekleme anlamına gelir.
- Yukarıda durumlar değilse, ekleme noktasını bulana kadar indeks dosyasına ikili arama uygulanacaktır.

Kayıt Bul

searchIndexes(int num, int *searchFlag, int *searchedptr)

1. Kullanıcıdan öğrenci numarasını al.
2. SearchIndexes() fonksiyonu kullanarak numara olup olmadığını kontrol etmek
3. searchFlag 1 ise, numra mevcut demektir. 0 ise yok demektir.
4. seachedptr parametresiyle indeks dosyasındaki satırın işaretçisini bilmek.
5. listSearchedIndex() kullanarak bir öğrencinin bütün kayıtları ekrana göstermek.
6. bu fonksiyonda, istenen öğrenci numarasına sahip ilk kaydı bulana kadar dizin dosyasında satır satır geri gidecektir.
7. bu öğrenci numarasının kayıtlarını bitirene kadar dizilere kayıt kayıt ekleyecek.(records[], indexes[])
8. bu fonksiyon, bir öğrenci numarasının kayıtlarının sayısını döndürür.

- index dizisi (indexes[]) kullanarak binary dosyasından tüm kayıtlar göstermek
- Kayıtlar kullanıcıya geri yazdırılacaktır.

```
-----Kayıt Bulmak-----
Enter Student Number(ogrenci no): 9
[1]
Student Number: 9
Class Code: 25
Student Grade: 84
-----
[2]
Student Number: 9
Class Code: 29
Student Grade: 95
-----
[3]
Student Number: 9
Class Code: 12
Student Grade: 69
-----
Çıkış için 0'e basın.
Ana menu dönmek için 1'e basın:
```

Kayıt Silmek veya Güncellemek

- kayıtSil() , kayıtGuncel() Fonksiyonlar
- 1. kayıtbul fonksiyon gibi öğrencinin tüm kayıtları bulup ekrana yazdırır.
- 2. kullanıcıya o öğrenci numarasından hangi kaydın silineceği veya güncelleneceği sorulacaktır.
- 3. Kayıtlar veri dosyasından geçici bir dosyaya kopyala.
- 4. Kayıtın indeksi ulaştığında
- 5. Bu kaydı oku ve puanını güncelle, sonra geçici dosyaya ekle.(Güncellemek)
- 6. Silmek için sadece oku ama geçici dosyasına kopyalamayı atla.(Silmek).
- 7. Dosyanın sonuna kopyalamaya devam et.
- 8. Önceki veri dosyasını kaldır.
- 9. Geçici dosyayı yeni veri dosyasıyla yeniden adlandır

```
-----Kayıt Guncel-----
Enter Student Number(ogrenci no): 9
[1]
Student Number: 9
Class Code: 25
Student Grade: 84
-----
[2]
Student Number: 9
Class Code: 29
Student Grade: 95
-----
Which one to update?[1:2] 1
New Score(Yeni Puan): 89
Updated

Çıkış için 0'e basın.
Ana menu dönmek için 1'e basın:
```

```
-----Kayıt Silmek-----
Enter Student Number(ogrenci no): 9
[1]
Student Number: 9
Class Code: 25
Student Grade: 84
-----
[2]
Student Number: 9
Class Code: 25
Student Grade: 666
-----
[3]
Student Number: 9
Class Code: 29
Student Grade: 95
-----
Which one to delete?[1:3] 2
Deleted.

Çıkış için 0'e basın.
Ana menu dönmek için 1'e basın:
```

Kayıt Ekleme

ilk olarak kullanıcıdan öğrenci no, ders kodu ve puan almak.

1. Veri Struct'a aktar.
2. fseek() kullanarak veri dosyanın son indeksine kadar ulaşın ve bir değişkenine kaydedin (“index” olsun)
3. Kullanıcıdan aldığım verileri veri dosyasına kaydet.
4. Ekleme noktasını bilmek için öğrenci numarasını ve index değerlerini searchindex() Fonksiyona iletin.
5. Index dosyasından geçici bir dosyaya satır satır kopyala ekleme noktasına kadar.
6. Yeni öğrenci numarası ve index bir satırda index dosyasına ekle.
7. Dosyanın sonuna kopyalamaya devam et.
8. Önceki index dosyasını kaldır.
9. Geçici dosyayı yeni index dosyasıyla yeniden adlandır

```
-----Kayıt Ekleme-----
Input Student Number(orenci No): 7
Input Class Code(ders kodu): 23
Input Student Score(Puan): 5

-----Saved-----

Çıkış için 0'e basın.
Ana menu dönmek için 1'e basın:
```

Ana Menu

```
-----Ana Menu-----
1->Kayıt Ekleme için -> 1
2->Kayıt Bulme için -> 2
3->Kayıt Silme için -> 3
4->Kayıt Güncelleme için -> 4
5->Tüm Kayıtlar Listele -> 5
6->Index Dosyası Göstermek -> 6
7->Index Dosyası Silmek -> 7
8->Index Dosyası Oluşturmak -> 8
0->Çıkış için -> 0 basınız:
```

Tüm kayıtlar göstermek

```
-----
Student Number(örenci No): 19
Class Code(Ders Kodu): 19
Student Score(Puan): 99
-----
Student Number(örenci No): 18
Class Code(Ders Kodu): 36
Student Score(Puan): 92
-----
Student Number(örenci No): 16
Class Code(Ders Kodu): 29
Student Score(Puan): 48
-----
Student Number(örenci No): 8
Class Code(Ders Kodu): 66
Student Score(Puan): 99
-----
Student Number(örenci No): 7
Class Code(Ders Kodu): 23
Student Score(Puan): 5
-----
Student Number(örenci No): 9
Class Code(Ders Kodu): 12
Student Score(Puan): 99
-----
Çıkış için 0'e basın.
Ana menu dönmek için 1'e basın:
```

Index file sorted

```
index.txt - Notepad
File Edit Format View Help
1 350
1 0
1 12
2 24
2 36
2 374
2 362
3 48
3 386
3 60
4 72
4 84
5 96
5 410
5 506
5 108
6 120
6 132
6 422
7 144
7 156
7 615
8 168
8 494
8 180
8 603
9 192
9 627
9 398
10 204
Ln 11, Col 5 80% Windows (CRLF) UTF-8
```

Index Dosyası Silmek

```
-----Index File Silmek-----
Deleted
Çıkış için 0'e basın.
Ana menu dönmek için 1'e basın:
```

```
-----Index Dosyası Göstermek-----
File not found
Çıkış için 0'e basın.
Ana menu dönmek için 1'e basın:
```

```
-----Index File Oluştur-----
Index File oluşturuldu
Çıkış için 0'e basın.
Ana menu dönmek için 1'e basın:
```



