

PROGRAMLAMA LABORATUVARI 1

1. PROJE

Barış KAKILLI

Bilgisayar Mühendisliği Bölümü

Kocaeli Üniversitesi

baris_kakilli38@hotmail.com

Özet

Bu rapor Programlama Laboratuvarı - 1 dersi 1. proje için açıklamaya yöneliktir. Raporda projenin tanımı, çözüme yönelik yapılan araştırmalar, kullanılan yöntemler, proje hazırlanırken geliştirme ortamı ve kod bilgisi gibi programın oluşumunu açıklayan başlıklara yer verilmiştir. Rapor sonunda projemi hazırlarken kullandığım kaynaklar bulunmaktadır.

1. Proje Tanımı

Projenin amacı C dili kullanarak dosyalama fonksiyonlarının pratiğinin yapılmasıdır. Bu amaçla ikili dosyaların ve metin dosyalarının kullanıldığı bir uygulama geliştirilecektir.

Bu proje için yoğun indeks yapısı gerçekleştirilecektir.

İkili dosya içinde öğrenci kayıtları, metin dosyası içinde öğrencinin numarası ve ikili dosya(veri dosyası) içindeki adresi yer alacak.

1.1 Projenin Özellikleri

A - İndeks dosyası oluşturma:

İndeks dosyası oluşturma fonksiyonu eğer indeks dosyası yoksa veri dosyasını indeksleyip oluşturuyor. Eğer indeks dosyası mevcutsa bu fonksiyon hata mesajı gösterir.

```
index dosyasi olusturuldu.  
index dosyasi siralaniyor. index dosyasi zaten mevcut.
```

B - Kayıt ekle:

Kayıt ekle fonksiyonu veri dosyasının sonuna kullanıcının girdiği değerler doğrultusunda kayıt ekler ve indeks dosyasını yeni kayıtla güncelleyip sıralar.

```
kaydi eklenecek ogrencinin numarası: 200201012  
kaydi eklenecek ders kodu: 101  
kaydi eklenecek dersin puanı: 100  
index dosyasi siralaniyor.
```

C - Kayıt bul:

Kullanıcıdan alınan öğrenci numarasına ait veri olup olmadığını kontrol eder eğer varsa o öğrencinin tüm ders kodu ve puanlarını gösterir.

```
bulmak istediginiz ogrencinin numarası:200201012  
ogrenciye ait birden fazla veri var.  
1-200201012 450 95  
2-200201012 665 90  
3-200201012 101 100
```

D - Kayıt sil:

Kullanıcıdan silmek için alınan öğrenci numarasına ait veri olup olmadığını kontrol eder eğer birden fazla veri varsa kullanıcının hangisini silmek istediğini sorar. Veriyi hem veri dosyasından hem de indeks dosyasından siler.

```
kaydini silmek istediginiz ogrencinin numarası:200201012
ogrenciye ait birden fazla veri var.
1-200201012 450 95
2-200201012 665 90
3-200201012 101 100
ogrenciye ait hangi kaydı silmek istiyorsunuz:3
3 nolu kayıt silinicek emin misiniz?
1-Evet
2-Hayır
1
veri başarıyla silindi.
index dosyası sıralanıyor.
```

E - Kayıt güncelle:

Kullanıcıdan güncellemek için alınan öğrenci numarasına ait veri olup olmadığını kontrol eder eğer birden fazla veri varsa kullanıcının hangisini güncellemek istediğini sorar. Veriyi hem veri dosyasında hem de indeks dosyasında günceller.

F - Veri dosyasını göster:

İkili dosyayı(veri dosyası) ekranda gösterir.

```
veri dosyasi gosteriliyor:
ogrenci no - ders kodu - puan
1-200201094 645 90
2-200201056 110 70
3-200201012 450 95
4-200201078 331 99
5-200201017 101 95
6-200201007 101 85
7-190102012 480 75
8-200201003 970 85
9-200201012 665 90
10-200202002 458 92
11-200202056 453 72
12-200202017 555 55
13-200202017 101 80
```

G - İndeks dosyasını göster:

metin dosyasını(indeks dosyası) ekranda gösterir.

```
index dosyasi gosteriliyor:
ogrenci no - veri dosyasindaki adresi
190102012 108
200201003 126
200201007 90
200201012 36
200201012 144
200201017 72
200201056 18
200201078 54
200201094 0
200202002 162
200202017 198
200202017 216
200202056 180
```

H - İndeks dosyasını sil:

indeks dosyasını disk üzerinden siler eğer indeks dosyası mevcut değilse hata mesajı gösterir.

```
Index dosyasi basariyla silindi.
```

```
Hata! Index dosyasi silinemedi
Ipucu: Index dosyasi olusturulmamis olabilir.
```

2 - Araştırmalar ve Yöntem

Projede veri ve indeks dosyasına ihtiyacım olduğu zaman boyutunu alıp işlem yaptım böylelikle gerekmediği zamanlarda dizi içinde veri tutmuyorum ve bellekten tasarruf ediyorum.

```
while(fscanf(fp, "%d", &low) == 1)
    verisayisi++;
verisayisi = verisayisi / 2;
```

veri sayısını ikiye bölme sebebim indeks dosyasında verilerin öğrenci numarası+adresi şeklinde tutulmasından ötürü veri sayısının yarısı kadar veri bulunuyor.

kayıt sil ve kayıt güncelle fonksiyonları birbirlerine çok benzer olduğu için ikisinin de aynı fonksiyon içerisinde yazdım. kayıt sil fonksiyonunu kullanmak için bir(1) kayıt güncelle fonksiyonunu kullanmak için iki(2) değerini yolluyorum .

```
if(sil_gunc == 1)
    printf("kaydini silmek istediginiz ogrencinin numarası:");
else
    printf("kaydini degistirmek istediginiz ogrencinin numarası:");
```

```
if(sil_gunc == 1)
{
    if(sil_gunc == 2)
```

Kayıt bulma işlemi için ikili arama(binary search) kullandım.

İkili arama yöntemi ile kayıt bulma işleminde eğer öğrenci numarası için birden fazla kayıt varsa ilk bulunduğu kaydı gösteriyordu bu sorunu düzeltmek için kaydın üst ve alt kısmında taramalar yaparak o öğrenci numarası için tüm verilere ulaştım.

orta(mid) değeri istenen veriye ulaşıncaya kadar adet ve midtest isimli 2 adet int oluşturup(adet veriden kaç adet olduğunu saymam için, midtest mid değerini kaybetmeden sayılar üzerinde test yapmak için) mid değerini midteste atıyorum. midtest değerinin maxa eşit olmadığı durumlarda(midtest değeri her zaman maxdan az olacak) adet ve midtesti sırasıyla arttırıyorum böylelikle yukarı doğru bir arama yapmış oluyorum.

ilk döngüden çıktıktan sonra mid değerini tekrardan midteste aktarıp bu sefer midtest değeri sıfırdan büyükse döngüye giriyor ve aşağı doğru bir arama yapmış oluyorum.

örnek:

öğrenci no - adres

```
...
3-190102012 108
4-200201003 126
5-200201012 144
6-200201012 234
7-200201012 253
8-200201012 271
9-200201017 72
10-200201045 289
...
```

binary search algoritmasının altı çizili değeri bulunduğunu varsayalım üstünde de değerler (7 ve üstü) olduğundan max değeri bizim bulduğumuz değerden yukarıda olmuş oluyor. O yüzden döngüye giriyoruz adet ve midtesti sırasıyla arttırarak üstte ekstra 2 tane daha aynı değer olduğunu buluyoruz. Adet değerimiz 3 olmuş oluyor.

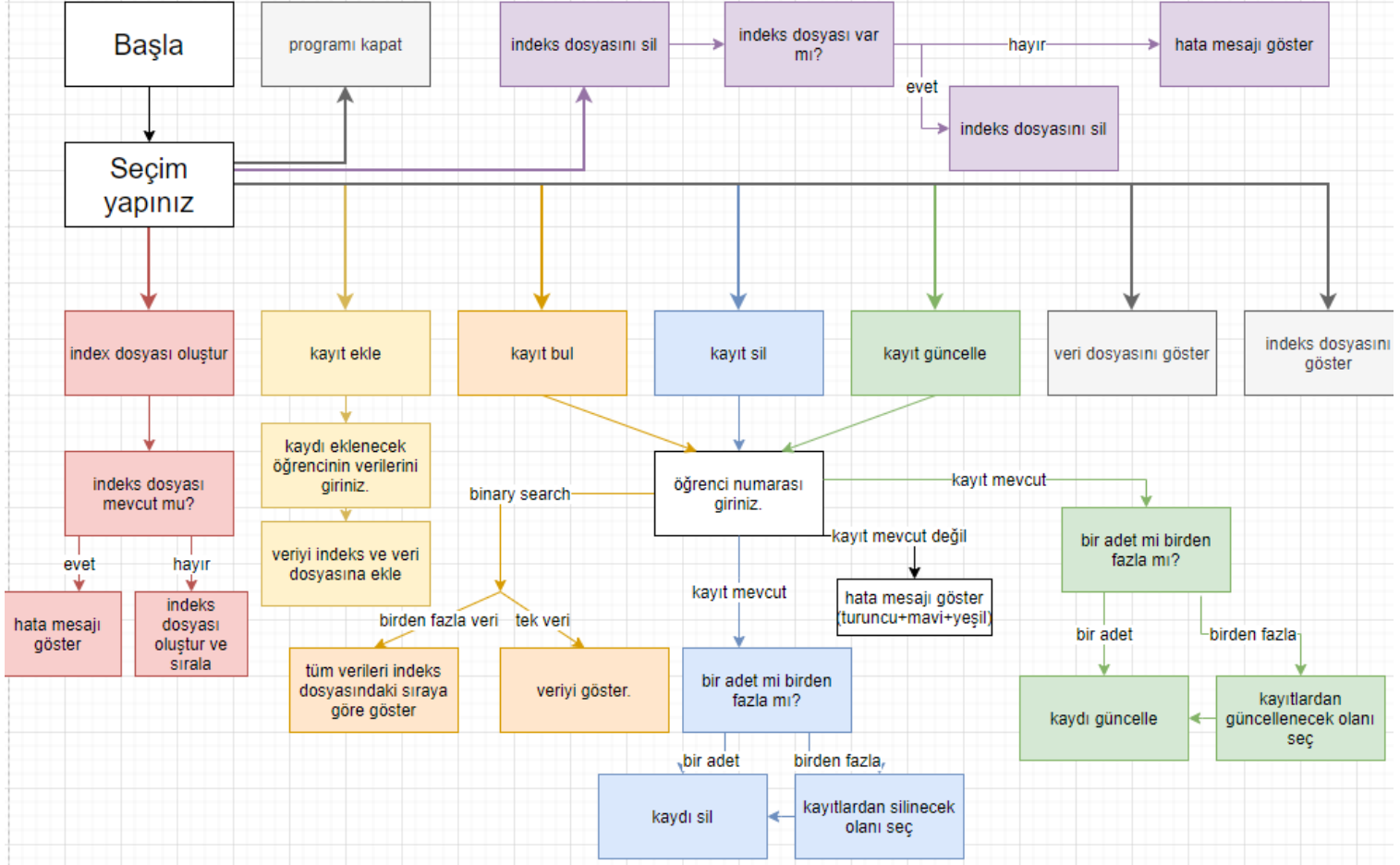
Sonra altı çizili değerimizin altında da değerler (5 ve altı) olduğu için döngüye giriyor ve alttaki 1 adet daha ekstra aynı değeri almış oluyoruz.

```
if(arr[mid] == istenen)
{
    int adet = 1, midtest;
    midtest = mid;
    if(midtest != max)
        while(arr[midtest] == arr[midtest+1])
        {
            adet++;
            midtest++;
        }
    midtest = mid;
    if(midtest > 0)
        while(arr[midtest] == arr[midtest-1])
        {
            adet++;
            midtest--;
        }
}
```

3- Geliştirme Ortamı

projeyi codeblocks 20.03 versiyonunda geliştirdim projeyi tek başıma yaptığım için paylaşma için ekstra bir sistem kullanmadım.

4 - Kod Bilgisi



Akış diyagramı

Kaynakça

- karşılaştığım çeşitli problemler:
<https://stackoverflow.com/>
- C dili dosya yönetimi:
<https://fresh2refresh.com/c-programming/c-file-handling/>
- Akış diyagramı oluşturmak için:
<https://app.diagrams.net/>