

# ALGORİTMALAR VE PROGRAMLAMA II

## Hafta 11 Dosya İşlemleri Sıralı Erişimli Dosyalar

# • Dosya İşlemleri

- ▶ Değişken ve dizilerde tutulan değerler programın çalışması esnasında değerlerini korurlarken programın çalışması sona erdiğinde veri kaybolur.
- ▶ Dosyalar verinin kalıcı olarak saklanması için kullanılır.
- ▶ Bilgisayarlar dosyaları disk, flash bellek gibi depolama aygıtlarında tutarlar.



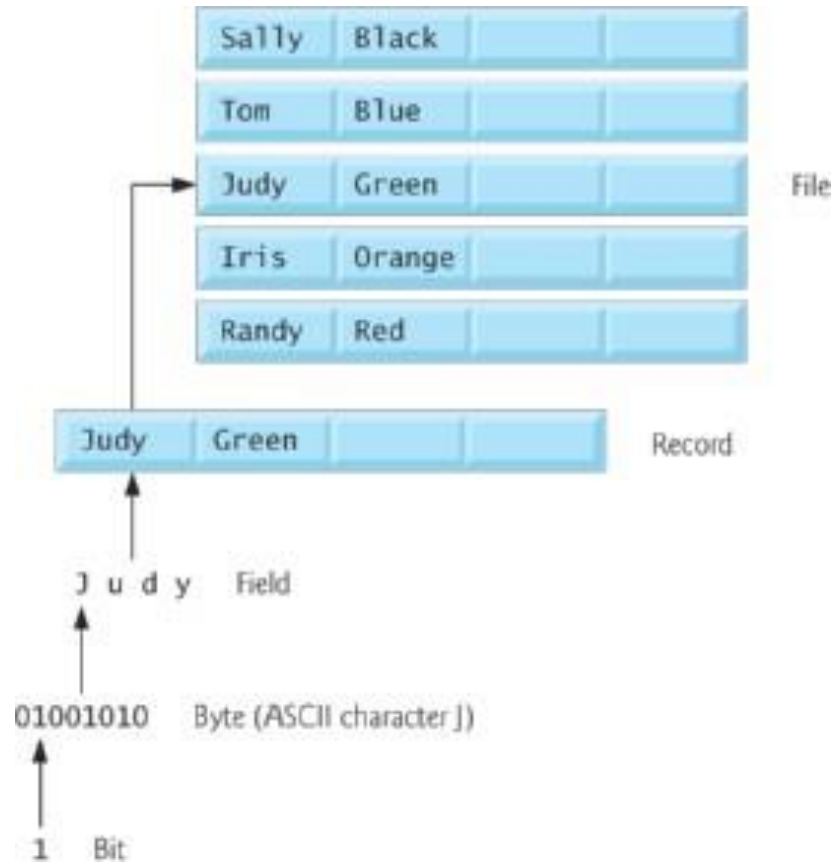
# • Veri Hiyerarşisi

Bilgisayarlar tarafından işlenen tüm veriler 1 ler ve 0 ların kombinasyonundan oluşur.

- **Bit:** Bilgisayar sistemlerindeki en küçük veri birimidir. En küçük birim olan bit değeri 1 veya 0 olabilir.
- **Byte:** Harfler, rakamlar, özel semboller karakterlerdir. Bilgisayarlar yalnızca 1 ler ve 0 lar ile işlem yapabildiği için bilgisayarlardaki karakter setinde yer alan her bir karakter 1 ler ve 0 ların diziliminden oluşur (bayt). 1 bayt = 8 bit
- **Alan:** Karakterlerin bir araya gelmesiyle oluşturduğu anlamlı bütün.  
Örn: kişi adı
- **Kayıt:** Birbiri ile ilişkili alanların bir araya gelmesiyle oluşur.
  - Struct veya class ile oluşturulur.
  - Örn: Bir okuldaki bir öğrenciye ait numara, ad, soyad, adres gibi alanların bir araya gelmesiyle bir kayıt oluşturulur.
- **Dosya:** Birbiri ile ilişkili kayıtların bir araya gelmesiyle oluşur.
  - Örn: Öğrencilere ait dosya
- **Database:** Birbiri ile ilişkili dosyaların bir araya getirilmesi ile oluşturulur.



# • Veri Hiyerarşisi



# • Veri Hiyerarşisi

- ▶ **Kayıt Anahtarı:** Bir dosya içerisindeki belli kayıtların elde edilebilmesi için en azından kayıtlara ait bir alanın anahtar olarak kullanılması gerekir.
  - Örn: Bir okul dosyasında öğrenci numarası kayıt anahtarı olarak belirlenebilir.
- ▶ **Sıralı Dosya:** Kayıtları dosya içerisinde tutmanın en popüler yöntemi.
  - Kayıtlar genellikle kayıt anahtarına göre sıralanır.



# Dosya ve Stream

- ▶ C’de bir dosya bayt dizisi olarak ele alınır.
- ▶ Dosya sonunda end-of-file işaretçisi bulunur veya dosya belirli bir bayt değerinde son bulur.
- ▶ Dosya açıldığında bir stream oluşturulur. Stream, dosya ile program arasında bir iletişim kanalı oluşturur.
- ▶ Bir dosyayı açmak dosyanın veri yapısını işaret eden bir pointer döndürür.
- ▶ Örnek dosya işaretçileri:
  - stdin - standard input (klavyeden veri okumayı sağlar)
  - stdout - standard output (ekrana veri yazdırmayı sağlar)
  - stderr - standard error (ekran)



# Dosya ve Stream

- ▶ **Dosya yapısı:** Bir dosya açıldığında dosyanın yapısını işaret eden bir pointer döndürülür.
- ▶ **Dosya tanımlayıcı:** İşletim sisteminin açık dosya tablosunda bir indistir.
- ▶ **Dosya Kontrol Bloğu (FCB):** Her dizi elemanında bulunur. Sistem bunu kullanarak dosyayı yönetir.

Giriş, çıkış ve hata işlemleri stdin, stdout ve stderr işaretçileri ile değiştirilir.



# • Dosya ve Stream

Oku/yaz fonksiyonları standart giriş kütüphanesinde bulunur.

## ▶ fgetc

- Dosyadan bir karakter okur
- Parametre olarak bir dosya işaretçisi alır
- `fgetc( stdin )` `getchar` ile aynı işi yapar

## ▶ fputc

- Dosyaya bir karakter yazar.
- Parametre olarak bir dosya işaretçisi ve yazılacak olan karakteri alır
- `fputc( 'a', stdout )`, `putchar( 'a' )` ile aynı işi yapar.

## ▶ fgets

- Dosyadan bir satır okur

## ▶ fputs

- Dosyaya bir satır yazar

## ▶ fscanf / fprintf

- `scanf` ve `printf` fonksiyonlarının dosya işlemlerindeki karşılıklarıdır





# Sıralı Erişimli Dosya Oluşturma

- ▶ C herhangi bir veri yapısını kullanmayı zorunlu kılmaz.
- ▶ Programcı kendi veri yapısını kendisi belirler.
- ▶ Dosya oluşturma:
  - ❑ **FILE \*myPtr;**
    - myPtr isminde bir dosya işaretçisi tanımlar.
  - ❑ **myPtr = fopen("myFile.dat", openmode);**
    - Fopen fonksiyonu belirtilen dosyayı gösteren bir dosya işaretçisi döndürür.
    - İki parametre alır – açılacak dosya ve dosyanın açılış modu.
    - Eğer dosya açma başarısız olursa NULL döner.
  - ❑ **fprintf**
    - Dosyaya yazmak için kullanılır.
    - Printf ten farkı ilk parametre yazılacak dosyayı işaret eden bir pointer dır.

# • Sıralı Erişimli Dosya Oluşturma

- ▶ feof( Dosya işaretçisi )
  - Eğer end-of-file göstergesi belirtilen dosya için ayarlanmış ise true döndürür.
- ▶ fclose( Dosya işaretçisi)
  - Belirtilen dosyayı kapatır.
  - Program sonlandırıldığında otomatik olarak gerçekleştirilir.
- ▶ Detaylar
  - Programlar bir veya daha fazla dosya üzerinde işlem gerçekleştirebilir.
  - Her dosyanın farklı bir ismi olmalı her birinin kendine özel işaretçisi olmalıdır.



# • Sıralı Erişimli Dosya Oluşturma

## ► Dosya açma modları

Mode	Description
<b>r</b>	Open a file for reading.
<b>w</b>	Create a file for writing. If the file already exists, discard the current contents.
<b>a</b>	Append; open or create a file for writing at end of file.
<b>r+</b>	Open a file for update (reading and writing).
<b>w+</b>	Create a file for update. If the file already exists, discard the current contents.
<b>a+</b>	Append; open or create a file for update; writing is done at the end of the file.



# Sıralı Erişimli Dosyaya Yazma

```
1 #include <stdio.h>
2
3 int main(void)
4 {
5     int hesapNo;
6     char ad[30];
7     double bakiye;
8     FILE *mfPtr; // musteri.dat dosyasi işaretcisi
9     if((mfPtr = fopen("musteri.dat","w")) == NULL)
10         printf("Dosya acilamadi\n");
11     else
12     {
13         printf("Hesap no, isim ve bakiye girin \n");
14         printf("Veri girisini bitirmek icin EOF gir"); //EOF = Ctrl + z
15         printf("? ");
16         scanf("%d%s%lf",&hesapNo,ad,&bakiye);
17
18         while(!feof(stdin))
19         {
20             fprintf(mfPtr,"%d %s %.2f \n",
21                 hesapNo,ad,bakiye);
22             printf("? ");
23             scanf("%d%s%lf",&hesapNo,ad,&bakiye);
24         }
25
26         fclose(mfPtr);
27     }
28     return 0;
29 }
```

# • Sıralı Erişimli Dosyaya Yazma

```
Enter the account, name, and balance.  
Enter EOF to end input.  
? 100 Jones 24.98  
? 200 Doe 345.67  
? 300 White 0.00  
? 400 Stone -42.16  
? 500 Rich 224.62  
? ^Z
```

# Sıralı Erişimli Dosyadan Okuma

- ▶ Bir dosya işaretçisi oluştur, dosyayı okumak için bu işaretçiyi dosyaya bağla
  - `myPtr = fopen( "myFile.dat", "r" );`
- ▶ **fscanf** kullanarak dosyadan oku
  - Scanf gibi ancak ilk parametre dosyayı gösteren bir işaretçidir
  - `fscanf( myPtr, "%d%s%f", &myInt, myString, &myFloat );`
- ▶ Baştan sona kadar dosya okunur
- ▶ Dosya konumu işaretçisi
  - Okunacak veya yazılacak sonraki bayt sayısını belirtir.
  - İşaretçi değildir, ancak bir tamsayıdır ( bayt konumunu belirten bir sayı)
  - Bayt ofseti de denir
- ▶ `rewind( myPtr )`
  - Dosya konumu işaretçisini dosyanın başına alır (bayt 0)



# Sıralı Erişimli Dosyadan Okuma

```
1 #include <stdio.h>
2
3 int main(void)
4 {
5     int hesapNo;
6     char ad[40];
7     double bakiye;
8     FILE *mfPtr; // musteriler.dat dosyası işareti
9     if((mfPtr = fopen("musteriler.dat","r")) == NULL)
10         printf("Dosya açılmadı\n");
11     else
12     {
13         printf("%-10s%-13s\n", "HesapNo","Ad","Bakiye");
14         fscanf(mfPtr,"%d%s%lf",&hesapNo,ad,&bakiye);
15
16         while(!feof(mfPtr))
17         {
18             printf("%-10d%-13s%7.2f\n", hesapNo,ad,bakiye);
19             fscanf(mfPtr,"%d%s%lf",&hesapNo,ad,&bakiye);
20         }
21         fclose(mfPtr);
22     }
23     return 0;
24 }
```

# Örnek Uygulama

```
1 #include <stdio.h>
2
3 int main(void)
4 {
5     int secim, hesapNo;
6     double bakiye;
7     char ad[40];
8     FILE *mfPtr;
9     if((mfPtr = fopen("musteri.dat","r")) == NULL)
10         printf("Dosya acilamadi\n");
11     else
12     {
13         printf("Secim yapiniz\n"
14             "1-Hesapta para olmayan hesaplar\n"
15             "2-Borclu olan hesaplar\n"
16             "3-Hesapta para olan hesaplar\n"
17             "4-Cikis\n");
18         scanf("%d",&secim);
```



# Örnek Uygulama

```
19     while(secim !=4)
20     {
21         fscanf(mfPtr,"%d%s%lf",&hesapNo,ad,&bakiye);
22         switch(secim)
23         {
24             case 1:
25                 printf("\nPara olmayan hesaplar :\n");
26                 while(!feof(mfPtr))
27                 {
28                     if(bakiye==0)
29                         printf("%-10d%-13s%7.2f\n", hesapNo,ad,bakiye);
30                     fscanf(mfPtr,"%d%s%lf",&hesapNo,ad,&bakiye);
31                 }
32                 break;
33             case 2:
34                 printf("\nBorclu hesaplar :\n");
35                 while(!feof(mfPtr))
36                 {
37                     if(bakiye<0)
38                         printf("%-10d%-13s%7.2f\n", hesapNo,ad,bakiye);
39                     fscanf(mfPtr,"%d%s%lf",&hesapNo,ad,&bakiye);
40                 }
41                 break;
```

# Örnek Uygulama

```
42         case 3:
43             printf("\nPara olan hesaplar :\n");
44             while(!feof(mfPtr))
45             {
46                 if(bakiye>0)
47                     printf("%-10d%-13s%7.2f\n", hesapNo,ad,bakiye);
48                 fscanf(mfPtr,"%d%s%lf",&hesapNo,ad,&bakiye);
49             }
50             break;
51         }
52         rewind(mfPtr);
53         printf("\n?");
54         scanf("%d",&secim);
55     }
56     printf("Program sonlandi\n");
57     fclose(mfPtr);
58 }
59 }
```

# Örnek Uygulama

```
Enter request
1 - List accounts with zero balances
2 - List accounts with credit balances
3 - List accounts with debit balances
4 - End of run
? 1

Accounts with zero balances:
300      White      0.00

? 2

Accounts with credit balances:
400      Stone     -42.16

? 3

Accounts with debit balances:
100      Jones      24.98
200      Doe       345.67
500      Rich      224.62

? 4
End of run.
```

# Sıralı Erişimli Dosyada Risk

- ▶ Sıralı erişimli dosya
  - Bir veriyi değiştirirken diğer verileri değiştirme riski taşır
  - Alanlar boyut olarak farklı olabilirler
    - ❑ 1, 34, -890 tümü tamsayıdır ancak disk üzerinde farklı boyutlardırlar
- ▶ 300 White 0.00 400 Jones 32.87 (dosyadaki eski veri)
- ▶ Eğer biz White ismini Worthington olarak değiştirmek istersek Jones kaydına ait veriler üzerine yeni veriler yazılır.

300 Worthington 0.00



300 White 0.00 400 Jones 32.87



300 Worthington 0.00 Jones 32.87

**Veri üzerine yazılır**



# Kaynaklar

**Doç. Dr. Caner ÖZCAN, KBÜ Yazılım Mühendisliği [www.canerozcan.net](http://www.canerozcan.net)\***

- ▶ Doç. Dr. Fahri Vatansever, “Algoritma Geliştirme ve Programlamaya Giriş”, Seçkin Yayıncılık, 12. Baskı, 2015.
- ▶ Kaan Aslan, “A’dan Z’ye C Klavuzu 8. Basım”, Pusula Yayıncılık, 2002.
- ▶ Paul J. Deitel, “C How to Program”, Harvey Deitel.
- ▶ “A book on C”, All Kelley, İra Pohl

\* Bu dersin slaytları genelde bu kaynaktan türetilmiştir.



# hru-algpro.github.io

Ders içerikleri ve duyurular için dersin web sitesine yukarıdaki adresten ulaşabilirsiniz.

S o r u l a r

?



Dinlediğiniz için teşekkürler

