

# BLM 112- Programlama Dilleri II

## Hafta 11

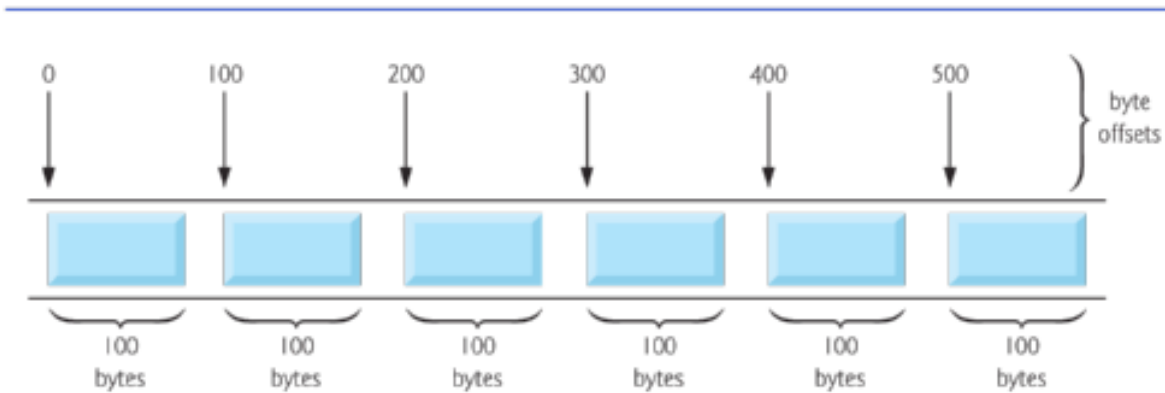
### Dosya İşlemleri Rasgele Erişimli Dosyalar

Dr. Öğr. Üyesi Caner Özcan

*Bir insanı değerlendirmek için nelere sahip olmadığına değil, sahip olduklarıyla neler yaptığına bak.*

# Rasgele Eriřimli Dosyalar

- ▶ Rasgele eriřimli dosyalar
  - İstenilen kayda dięer kayıtlar üzerinde arama yapmadan eriřim.
  - Dosya içindeki kayıtlara anlık eriřim
  - Yeni veri mevcut veriyi bozmadan dosyaya eklenebilir
  - Daha önce saklanan kayıt üzerine yazılmadan güncellenebilir veya silinebilir
- ▶ Sabit kayıt uzunluęu kullanılarak uygulanır
  - Sıralı eriřimli dosyalarda sabit kayıt uzunluęu yoktur.



# Rasgele Erişimli Dosyalar

- ▶ Rasgele erişimli dosyadaki veri biçimlendirilmemiş ("raw bytes" olarak saklanır)
  - Aynı tipteki tüm veriler (örn. int tipi) aynı miktarda hafıza alanı kullanır.
  - Aynı tipteki kayıtlar aynı uzunluktadır
  - İçeriği insanlar tarafından okunabilir değildir



# Rasgele Erişimli Dosya Oluşturma

- ▶ Biçimlendirmesiz I/O fonksiyonları
  - **fwrite**
    - Hafızanın bir bölgesinden dosyaya bayt transferi yapar.
  - **fread**
    - Dosyadan hafızanın bir bölgesine bayt transferi yapar
- ▶ Örneğin:
  - **fwrite( &number, sizeof( int ), 1, myPtr );**
    - **&number** – Bayt'ların kendisinden transfer edileceği yer
    - **sizeof( int )** – Transfer edilecek bayt sayısı
    - **Örn: 1** girilirse diziler için dizinin bir elemanının transfer edilmesi
    - **myPtr** – Dosyayı gösteren pointer.

# Rasgele Erişimli Dosya Oluşturma

- ▶ Struct veri yapısını dosyaya yazdırma
  - `fwrite( &myObject, sizeof (struct myStruct), 1, myPtr );`
  - **sizeof** – nesnenin kapladığı hafıza miktarını gösterir
- ▶ Birden fazla dizi elemanı yazdırmak için
  - İlk parametre diziyi gösteren pointer olmalı
  - Üçüncü parametre de yazdırılacak eleman sayısını belirtmeli

# Rasgele Erişimli Dosya Oluşturma

```
1 #include <stdio.h>
2
3 struct musterisi{
4     int hesapNo;
5     char soyad[25];
6     char ad[20];
7     double bakiye;
8 };
9
10 int main(void)
11 {
12     int i;
13     struct musterisi bosMusterisi = {0, "", "", 0.0};
14     FILE *myPtr;
15     if((myPtr = fopen("musterisi.dat", "w")) == NULL)
16         printf("Dosya olusturulamadi\n");
17     else
18     {
19         for(i=1; i<=100; i++)
20         {
21             fwrite(&bosMusterisi, sizeof(struct musterisi), 1, myPtr);
22         }
23         fclose(myPtr);
24     }
25     return 0;
26 }
```

# Rasgele Erişimli Dosyaya Yazma

- ▶ **fseek** : Dosyadaki konumu belirten işaretçinin konumunu ayarlar.
- ▶ **fseek( pointer, offset, symbolic\_constant );**
  - ❑ **pointer** – dosyayı gösteren işaretçi
  - ❑ **offset** – dosyadaki konumu gösteren işaretçi (ilk pozisyon 0)
  - ❑ **symbolic\_constant** – dosyanın neresinden okumaya başlayacağımızı gösterir.
    - **SEEK\_SET** – dosyanın başından başla
    - **SEEK\_CUR** – dosyanın mevcut bulunulan konumundan başla
    - **SEEK\_END** – dosyanın sonundan başla





# Rasgele Erişimli Dosyaya Yazma

```
26
27     fseek(myPtr,(hesapBilgi.hesapNo-1)*
28           sizeof(struct musteriler),SEEK_SET);
29
30     fwrite(&hesapBilgi,sizeof(struct musteriler),1,myPtr);
31
32     printf("Hesap no gir\n?");
33     scanf("%d",&hesapBilgi.hesapNo);
34 }
35     fclose(myPtr);
36 }
37     return 0;
38 }
```

# Rasgele Erişimli Dosyadan Okuma

## ▶ fread

- Dosyadan belirtilen kadar bayt okuyup hafızaya atar

## ▶ `fread( &client, sizeof (struct clientData), 1,myPtr );`

- ❖ Birden fazla sabit uzunlukta dizi elemanı okuyabilir.
  - Diziyi gösteren işaretçi oluştur
  - Okunacak eleman sayısını belirt
- ❖ Birden çok eleman okumak için üçüncü parametrede belirt

# Rasgele Erişimli Dosyadan Okuma

```
1 #include <stdio.h>
2
3 struct musterisi{
4     int hesapNo;
5     char soyad[25];
6     char ad[20];
7     double bakiye;
8 };
9
10 int main(void)
11 {
12     struct musterisi hesapBilgi = {0, "", "", 0.0};
13     FILE *myPtr;
14     if((myPtr = fopen("musterisi.dat", "r"))== NULL)
15         printf("Dosya acilamadi\n");
16     else
17     {
18         printf("%-10s%-16s%-11s%10s\n", "HesapNo", "Soyad", "Ad", "Bakiye");
19         while(!feof(myPtr))
20         {
21             fread(&hesapBilgi, sizeof(struct musterisi), 1, myPtr);
22             if(hesapBilgi.hesapNo!=0)
23                 printf("%-10d%-16s%-11s%10.2f\n", hesapBilgi.hesapNo,
24                     hesapBilgi.soyad, hesapBilgi.ad, hesapBilgi.bakiye);
25         }
26         fclose(myPtr);
27     }
28     getchar();
29     return 0;
30 }
```

# Örnek Uygulama

- ▶ Bu uygulamada
  - Banka hesap bilgileri üzerinde çeşitli anlık işlemlerin yapılabilmesi için rasgele erişimli dosya kullanımı gösterilecektir.
- ▶ Yapılacak işlemler
  - Mevcut hesapları güncelle
  - Yeni hesap ekle
  - Mevcut bir hesabı sil
  - Dosyadaki hesap bilgilerini formatlı biçimde bir metin dosyasına yazdır.

# Örnek Uygulama

```
1 #include <stdio.h>
2
3 struct musteriy{
4     int hesapNo;
5     char soyad[25];
6     char ad[20];
7     double bakiye;
8 };
9 int secimGir(void);
10 void textDosya(FILE *);
11 void kayitGuncelle(FILE *);
12 void yeniKayit(FILE *);
13 void kayitSil(FILE *);
14 void listele(FILE *);
15
16 int main(void)
17 {
18     FILE *myPtr;
19     int secim;
20     if((myPtr = fopen("musteriy.dat","r+"))== NULL)
21         printf("Dosya acilamadi\n");
22     else
23     {
24         while((secim = secimGir()) != 6)
25         {
```

# Örnek Uygulama

```
26         switch(secim)
27         {
28             case 1:textDosya(myPtr);break;
29             case 2:kayitGuncelle(myPtr);break;
30             case 3:yeniKayit(myPtr);break;
31             case 4:kayitSil(myPtr);break;
32             case 5:listele(myPtr);break;
33         }
34     }
35     fclose(myPtr);
36 }
37 }
```

# Örnek Uygulama

```
39 int secimGir()
40 {
41     int menuSecim;
42     printf("\n Secimini yap\n"
43           "1-musteri.dat dosyasinin icerigini\n"
44           "  formatli olarak \"hesaplar.dat\" dosyasina yaz\n"
45           "2-hesap guncelle\n"
46           "3-yeni hesap ekle\n"
47           "4-hesap sil\n"
48           "5-musteri.dat dosyasinin icerigini listele\n"
49           "6-cikis\n?");
50     scanf("%d",&menuSecim);
51     return menuSecim;
52 }
```

# Örnek Uygulama

```
54 void textDosya(FILE *okuPtr)
55 {
56     FILE *yazPtr;
57     struct musteriler hesapBilgi = {0, "", "", 0.0};
58     if((yazPtr = fopen("hesaplar.dat", "w")) == NULL)
59         printf("Dosya acilamadi\n");
60     else
61     {
62         rewind(okuPtr);
63         fprintf(yazPtr, "%-10s%-16s%-11s%10s\n", "HesapNo", "Soyad", "Ad", "Bakiye");
64         while(!feof(okuPtr))
65         {
66             fread(&hesapBilgi, sizeof(struct musteriler), 1, okuPtr);
67             if(hesapBilgi.hesapNo != 0)
68                 fprintf(yazPtr, "%-10d%-16s%-11s%10.2f\n", hesapBilgi.hesapNo,
69                     hesapBilgi.soyad, hesapBilgi.ad, hesapBilgi.bakiye);
70         }
71         fclose(yazPtr);
72     }
73 }
```



# Örnek Uygulama

```
75 void kayitGuncelle(FILE *fPtr)
76 {
77     int hesapID;
78     double islemMiktari;
79     struct musteriler hesapBilgi = {0, "", "", 0.0};
80     printf("Guncellenecek hesap no gir[1-100]:");
81     scanf("%d",&hesapID);
82     fseek(fPtr,(hesapID-1)*sizeof(struct musteriler),SEEK_SET);
83     fread(&hesapBilgi,sizeof(struct musteriler),1,fPtr);
84     if(hesapBilgi.hesapNo==0)
85         printf("%d nolu hesap için bilgi girilmemiş\n",hesapID);
86     else
87     {
88         printf("%-10d%-16s%-11s%10.2f\n\n",hesapBilgi.hesapNo,
89             hesapBilgi.soyad,hesapBilgi.ad,hesapBilgi.bakiye);
90         printf("Hesaba yatacak (+) veya hesaptan çekilecek (-) tutari gir:");
91         scanf("%lf",&islemMiktari);
92         hesapBilgi.bakiye += islemMiktari;
93         printf("%-10d%-16s%-11s%10.2f\n\n",hesapBilgi.hesapNo,
94             hesapBilgi.soyad,hesapBilgi.ad,hesapBilgi.bakiye);
95         fseek(fPtr,(hesapID-1)*sizeof(struct musteriler),SEEK_SET);
96         fwrite(&hesapBilgi,sizeof(struct musteriler),1,fPtr);
97     }
98 }
```

# Örnek Uygulama

```
100 void kayitSil(FILE *fPtr)
101 {
102     struct musterisi hesapBilgi, bosHesap = {0, "", "", 0.0};
103     int hesapID;
104     printf("Silinecek hesap no gir[1-100]:");
105     scanf("%d",&hesapID);
106     fseek(fPtr,(hesapID-1)*sizeof(struct musterisi),SEEK_SET);
107     fread(&hesapBilgi,sizeof(struct musterisi),1,fPtr);
108     if(hesapBilgi.hesapNo==0)
109         printf("Silinecek %d nolu hesap yok",hesapID);
110     else
111     {
112         fseek(fPtr,(hesapID-1)*sizeof(struct musterisi),SEEK_SET);
113         fwrite(&bosHesap,sizeof(struct musterisi),1,fPtr);
114     }
115 }
```

# Örnek Uygulama

```
117 void yeniKayit(FILE *fPtr)
118 {
119     int hesapID;
120     struct musteriler hesapBilgi = {0, "", "", 0.0};
121     printf("Yeni hesap no gir[1-100]:");
122     scanf("%d",&hesapID);
123     fseek(fPtr,(hesapID-1)*sizeof(struct musteriler),SEEK_SET);
124     fread(&hesapBilgi,sizeof(struct musteriler),1,fPtr);
125     if(hesapBilgi.hesapNo!=0)
126         printf("%d nolu hesap zaten mevcut\n",hesapID);
127     else
128     {
129         printf("Soyad, Ad ve bakiye gir:");
130         scanf("%s%s%lf",hesapBilgi.soyad,hesapBilgi.ad,&hesapBilgi.bakiye);
131         hesapBilgi.hesapNo = hesapID;
132         fseek(fPtr,(hesapID-1)*sizeof(struct musteriler),SEEK_SET);
133         fwrite(&hesapBilgi,sizeof(struct musteriler),1,fPtr);
134     }
135 }
```

# Örnek Uygulama

```
137 void listele(FILE *fPtr)
138 {
139     struct musteriler hesapBilgi = {0, "", "", 0.0};
140
141     printf("%-10s%-16s%-11s%10s\n", "HesapNo", "Soyad", "Ad", "Bakiye");
142     while(!feof(fPtr))
143     {
144         fread(&hesapBilgi, sizeof(struct musteriler), 1, fPtr);
145         if(hesapBilgi.hesapNo != 0)
146             printf("%-10d%-16s%-11s%10.2f\n", hesapBilgi.hesapNo,
147                 hesapBilgi.soyad, hesapBilgi.ad, hesapBilgi.bakiye);
148     }
149     fclose(fPtr);
150     getchar();
151 }
```

# Ev Ödevi

## ► Hasta takip sistemi

- Hasta no, adı, yaşı ve hastalık bilgisini tutacak bir struct oluştur
- Belli sayıda hasta girişi yap.
- Dosya içerisinde hasta no'ya göre arama yap.
- Hasta kaydı sil
- Hasta kaydı düzenle.
- Dosyadan hastaları listele

# Gelecek Hafta

- ▶ Bitwise İşlemler



# Kaynaklar

- ▶ 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

S o r u l a r  
?



Dinlediğiniz için teşekkürler

CANER ÖZCAN



canerozcan@karabuk.edu.tr