

ALGORİTMA VE PROGRAMLAMA II

HAFTA#11

YZM 1106

Celal Bayar Üniversitesi Hasan Ferdi Turgutlu
Teknoloji Fakültesi

Genel Bakış...

2

- Dosya İşlemleri (Rastgele Erişim)
 - fread()
 - fwrite()
 - rewind()
 - fseek()
 - remove()

8. BÖLÜM

3

Dosya İşlemleri - Bölüm 3

fread() ve fwrite() Fonksiyonları

- Verileri dosyaya **yazma** ve dosyadan **okuma** işlemlerinde **fwrite()** ve **fread()** fonksiyonları da kullanılabilir.
- Daha önce kullandığımız **fprintf()** ve **fscanf()** fonksiyonları dosyadan veri okuma ve dosyaya veri yazma işlemlerinde **veriler üzerinde değişim yaparak çalıştığından**, **fread()** ve **fwrite()** fonksiyonlarını kullanmak daha pratiktir.

fread() ve fwrite() Fonksiyonları (devam...)

5

- `fprintf()` fonksiyonunu kullanarak bir dosyaya sayı (int) yazarken, **sayının dosyanın ASCII metnine çevrilmesi** gerekir.

```
fprintf(fp_ptr, "%d", sayi );
```

- Yine `fscanf()` fonksiyonu ile bir dosyadan bir sayı okurken, **sayının `fscanf()` fonksiyonunun dahili format yapısına çevrilmesi** gerekir.

```
fscanf (fp_ptr, "%d", &sayi);
```

fread() ve fwrite() Fonksiyonları (devam...)

6

```
size_t fread ( void *tampon-bellek,  
               size_t boyut,  
               size_t id1,  
               FILE *fp) ;
```

- **fp** ile gösterilen dosyadan okunan boyut yapısında **id1** kadar değeri **tampon-bellek** ile gösterilen belleğe atar.
- Burada, **boyut** ifadesi okunan verinin **byte** olarak değerini, **id1** ifadesi ise kaç adet veri okunduğunu belirler.
- **fread()** fonksiyonu okunan veri sayısını geri verir. Bu değer **0** ise, herhangi bir veri okunmamış demektir. Bu durumda, ya bir **hata olmuştur** ya da **dosya sonu gelmiştir**.

fread() ve fwrite() Fonksiyonları (devam...)

7

```
size_t fwrite (void *tampon-bellek,  
              size_t boyut,  
              size_t id1,  
              FILE *fp);
```

- **fwrite()** fonksiyonu, **fread()** fonksiyonunun yaptığı işlemin tam tersini yapar. **fwrite()** fonksiyonu, **tampon-bellek** ile gösterilen bellekte bulunan **boyut** yapısında **id1** kadar değeri **fp** ile gösterilen dosyaya yazar. Burada, **boyut** parametresi yazılan **byte** olarak değerini, **id1** parametresi ise kaç adet veri yazıldığını belirler.
- **fwrite()** fonksiyonu yazılan veri sayısını geri verir. Sadece bir hata meydana geldiğinde bu değer **id1** değerinden az olur.

Örnek : fread ve fwrite İlk Kullanımı

8

1. `int` veri türünde `a` isimli bir değişken tanımlayınız. Değişkenin değerine `5` atayınız.
2. `fwrite()` kullanarak değişkeni “deneme.txt” dosyasına yazdırınız. Dosyayı kapatınız.
3. Dosyayı tekrar açınız, `fread()` ile okuma yaparak, `int` veri türünde `b` isimli değişkene okuduğunuz değeri aktarınız.
4. `b` değişkeninin değerini ekrana yazdırınız.

Örnek : fread ve fwrite İlk Kullanımı

9

```
int main()
{
    FILE *fp;
    int a = 5;
    int b;

    fp = fopen( "deneme.txt" , "w" );
    fwrite(&a, sizeof(int), 1, fp);
    fclose(fp);

    fp = fopen( "deneme.txt" , "r" );
    fread(&b, sizeof(int), 1, fp);

    printf("b: %d", b);
    fclose(fp);

    return 0;
}
```

Örnek 1: fread ve fwrite Kullanımı

10

1. `int` veri türünde `dizi` isimli 5 elemanı bir dizi tanımlayınız.
2. Dizinin elemanları `klavyeden girilerek bir döngü içerisinde` doldurulacaktır.
3. Dizi değerleri `deneme.txt` isimli bir dosyaya `fwrite()` fonksiyonu kullanılarak yazılıp ve dosya kapatılacaktır.
4. Dizi elemanlarının değerlerine `bir döngü içerisinde 0 değeri atanıp` dizi elemanları `ekrana yazdırılacaktır`.
5. Daha sonra `deneme.txt` dosyasındaki dizi elemanları `fread()` fonksiyonu kullanılarak bir döngüye `diziye aktarılır`.
6. Dizi elemanları `ekrana yazdırılır`.

Örnek 1: fread ve fwrite Kullanımı

11

```
1. int degeri giriniz: 10
2. int degeri giriniz: 20
3. int degeri giriniz: 30
4. int degeri giriniz: 456
5. int degeri giriniz: 9011

Degerleri silinmis dizi elemanlari:
0 0 0 0 0
Dosyadan okunan dizi degerleri:
10 20 30 456 9011
```

Örnek 1: fread ve fwrite Kullanımı

12

1, 2 ve 3 nolu adımlar

```
int main (void)
{
    FILE *fp;
    int dizi[5];
    int i;

    for (i = 0; i < 5; i++)
    {
        printf("%d. int degeri giriniz: ", i + 1);
        scanf("%d", &dizi[i]);
    }

    if ((fp=fopen ("deneme.txt", "wb")) == NULL)
    {
        printf("Dosya açılmadı!\n");
        exit(1);
    }

    for (i = 0; i < 5; i++)
    {
        if (fwrite(&dizi[i], sizeof(int), 1, fp) != 1)
        {
            printf("Yazma hatası!\n");
            exit(1);
        }
    }
    fclose(fp);
}
```

Örnek 1: fread ve fwrite Kullanımı

13

4, 5 ve 6 nolu adımlar

```
if ((fp=fopen("deneme.txt", "rb")) == NULL)
{
    printf("Dosya açılmadı!\n");
    exit(1);
}

printf("\nDegerleri silinmis dizi elemanlari:\n");
for (i = 0; i < 5; i++)
{
    dizi[i] = 0;
    printf("%d ", dizi[i]);
}
printf("\n");

for (i = 0; i < 5; i++) {
    if (fread(&dizi[i], sizeof(int), 1, fp) != 1)
    {
        printf("Okuma hatası!\n");
        exit(1);
    }
}
fclose(fp);

printf("Dosyadan okunan dizi degerleri:\n");
for (i = 0; i < 5; i++)
    printf("%d ", dizi[i]);

return 0;
```

Örnek 1: fread ve fwrite Kullanımı - Eklenti

14

1. Diziden veriler dosyaya yazılırken ve dosyadan okunurken işlemleri döngü içerisinde değil tek seferde tamamlayalım.

```
if (fwrite(dizi, sizeof(dizi), 1, fp) != 1)
{
    printf("Yazma hatası!\n");
    exit(1);
}

if (fread(dizi, sizeof(dizi), 1, fp) != 1)
{
    printf("Okuma hatası!\n");
    exit(1);
}
```

rewind() Fonksiyonu

15

- Normalde bir dosyaya yazma işlemini tamamladıktan sonra, dosyayı tekrar okumak istediğimizde dosyayı kapatır ve dosyayı okuma modunda tekrar açarız.

```
fopen ("w" );  
dosyaYazmaIslemleri ();  
fclose (fp) ;  
fopen ("r" );  
dosyaOkumaIslemleri ();  
fclose (fp) ;
```

rewind() Fonksiyonu (devam...)

16

- Dosya açma kapatma işlemi yerine, bir dosyanın aktif konumunu dosya başına almak için `rewind()` fonksiyonunu kullanabilir.

```
void rewind (FILE *fp);
```

- `fp` parametresi ile gösterilen dosyanın aktif konum göstergesini dosyanın başına alır.
- Bu fonksiyon herhangi bir değer geri vermez. Çünkü, başarılı bir şekilde açılan dosyanın aktif konumu başa alınabilir.

Örnek 2: rewind ve struct Kullanımı

17

1. Aşağıdaki yapıyı tanımlayınız.

```
typedef struct  
{  
    int HesapNo;  
    char Isim[100];  
    float Bakiye;  
} stHesap;
```

2. `musteri.txt` isimli dosyaya **Hesap No değeri 0 olana kadar** Hesap bilgilerini yazılmasını sağlayın.
3. Dosyaya yazma işlemi için `fwrite()` fonksiyonunu kullanın.
4. Yazma işleminden sonra dosya başına `rewind()` fonksiyonu kullanarak konumlandırma yapın.
5. Son olarak dosyadan okuma işlemini `fread()` fonksiyonu kullanarak gerçekleştirin.

Örnek 2: rewind ve struct Kullanımı

18

1. Aşağıdaki yapıyı tanımlayınız.

```
typedef struct  
{  
    int HesapNo;  
    char Isim[100];  
    float Bakiye;  
} stHesap;
```

2. `musteri.txt` isimli dosyaya **Hesap No değeri 0 olana kadar** Hesap bilgilerini yazılmasını sağlayın.
3. Dosyaya yazma işlemi için `fwrite()` fonksiyonunu kullanın.
4. Yazma işleminden sonra dosya başına `rewind()` fonksiyonu kullanarak konumlandırma yapın.
5. Son olarak dosyadan okuma işlemini `fread()` fonksiyonu kullanarak gerçekleştirin.

Örnek 2: rewind ve struct Kullanımı

19

```
-----HESAP UERI GIRISI-----  
Hesap No:1000  
Isim:Deniz  
Bakiye:456910  
  
Hesap No:1001  
Isim:Ada  
Bakiye:345980  
  
Hesap No:0  
  
-----HESAP LISTESI-----  
Hesap No:1000  
Isim:Deniz  
Bakiye:456910.000000  
  
Hesap No:1001  
Isim:Ada  
Bakiye:345980.000000
```

Örnek 2: rewind ve struct Kullanımı

20

```
typedef struct
{
    int HesapNo;
    char Isim[100];
    float Bakiye;
} stHesap;

int main()
{
    FILE *fp;
    stHesap hesap;

    if ((fp=fopen ("musteri.txt", "wb+")) == NULL)
    {
        printf("Dosya açılmadı!\n");
        exit(1);
    }

    printf("\n-----HESAP VERI GIRISI-----\n");
    while (1 == 1)
    {
        printf("Hesap No:");
        scanf ("%d", &hesap.HesapNo);

        if (hesap.HesapNo == 0)
            break;
    }
}
```

Örnek 2: rewind ve struct Kullanımı

21

```
printf("Isim:");
scanf("%s", hesap.Isim);
printf("Bakiye:");
scanf("%f", &hesap.Bakiye);

if (fwrite(&hesap, sizeof(hesap), 1, fp) != 1)
{
    printf("Yazma hatası!\n");
    exit(1);
}
printf("\n");
rewind(fp);

stHesap hesap2;
printf("\n-----HESAP LISTESI-----\n");
while(fread(&hesap2, sizeof(hesap2), 1, fp) == 1)
{
    printf("Hesap No:%d\n", hesap2.HesapNo);
    printf("Isim:%s\n", hesap2.Isim);
    printf("Bakiye:%f\n", hesap2.Bakiye);
    printf("\n");
}

return 0;
}
```

fseek() Fonksiyonu

22

- Şimdiye kadar yazdığımız bütün kodlarda, dosyadan yaptığımız okuma işlemlerini dosyanın başından sonuna doğru bir sıra dahilinde yaptık.
 - Bu tipte dosya erişimine **Sıralı Erişim** denilmektedir.
- Bunun yanında, **fseek()** fonksiyonunu kullanarak dosyaların herhangi bir yerindeki bilgi okunabilir.
 - Bu tipte dosya erişimine **Rastgele Erişim** denilmektedir.

fseek() Fonksiyonu (devam...)

23

```
int fseek (FILE *fp, long ara, int yer);
```

- `fp` parametresi işlem yapılan dosyayı,
- `ara` parametresi `yer` parametresinin gösterdiği değerin tanımladığı dosya konumundan, işlem yapılmak istenen yerin `byte` olarak uzaklığını verir.
- `yer` parametresi dosyada arama işleminin başlayacağı yeri gösterir. **3 tane değeri vardır.**

fseek() Fonksiyonu (devam...)

24

- **yer** parametresinin alabileceği değerler aşağıdaki gibidir:
 - SEEK_SET (0) Aramayı dosya başından başlatır.
 - SEEK_CUR (1) Aramayı aktif konumdan başlatır.
 - SEEK_END (2) Aramayı dosya sonundan başlatır.

Örnek 3: Rastgele Erişimli Bir Dosyaya Rastgele Veri Yazmak

25

1. Aşağıdaki yapıyı tanımlayınız.

```
typedef struct
{
    int HesapNo;
    char Isim[100];
    float Bakiye;
} stHesap;
```

2. `musteri.txt` isimli dosyaya **Hesap No değeri 0 olana kadar** Hesap bilgilerinin yazılmasını sağlayın.
3. Hesap No'ya göre dosyanın konumlanmasını sağlayın. Örn: Hesap No: 4 girilirse ise kaydı `fseek()` fonksiyonu ile konumlandırıp, dosyada 4. sıraya ekleyelim.
4. Dosyaya yazma işlemi için `fwrite()` fonksiyonunu kullanın.
5. Yazma işleminden sonra dosya başına `rewind()` fonksiyonu kullanarak konumlandırma yapın.
6. Son olarak dosyadan okuma işlemini `fread()` fonksiyonu kullanarak gerçekleştirin.

Örnek 3: Rastgele Erişimli Bir Dosyaya Rastgele Veri Yazmak

26

```
-----HESAP UERI GIRISI (1-100 ARSI)-----  
Hesap No (1-100 arasi):1  
Isim:Deniz  
Bakiye:100  
  
Hesap No (1-100 arasi):4  
Isim:Ada  
Bakiye:200  
  
Hesap No (1-100 arasi):0  
  
-----HESAP LISTESI-----  
Hesap No:1  
Isim:Deniz  
Bakiye:100.000000  
  
Hesap No:0  
Isim:  
Bakiye:0.000000  
  
Hesap No:0  
Isim:  
Bakiye:0.000000  
  
Hesap No:4  
Isim:Ada  
Bakiye:200.000000
```

Örnek 3: Rastgele Erişimli Bir Dosyaya Rastgele Veri Yazmak

27

Yazma işleminden hemen önce
dosya konumlandırma işlemi **yapılmalıdır**.

```
fseek(fp, (hesap.HesapNo - 1) * sizeof(hesap), SEEK_SET);
```

remove() Fonksiyonu

28

- Bir dosyayı silmek için kullanılır.

```
int remove (char *dosya-adı);
```

- `remove()` fonksiyonu `dosya-adı` parametresi ile gösterilen dosyayı siler.
- Başarılı bir şekilde sona erdiğinde 0 değerini, aksi takdirde 0 olmayan bir değer geri verir.

Örnek 4: Dosya Silme

29

1. musterit.txt isimli dosya program tarafından oluşturulacaktır. Daha sonra aşağıdaki ekran çıktısı desteklenecektir.

```
Dosya olusturuldu.....  
Dosya silinecek! Devam edecek misiniz (E/H)?
```

Farklı Örnek: Bilgisayarı Kapatma

30

1. C sistem kaynaklarına direk erişebildiği için C ile farklı mümkündür.
2. Bu kapsamda kullanılan fonksiyonlardan bir tanesi **system** dir.

```
system("C:\\WINDOWS\\System32\\shutdown /s");
```

Farklı Örnek: Bilgisayarı Kapatma

31

```
#include <stdio.h>
#include <stdlib.h>

int main()
{
    printf("\nBilgisayar kapatılacak! Emin misiniz (E/H)?");
    if (toupper(getche()) == 'E')
        system("C:\\WINDOWS\\System32\\shutdown /s");

    return 0;
}
```

KAYNAKLAR

32

- N. Ercil Çağıltay ve ark., C DERSİ PROGRAMLAMAYA GİRİŞ, Ada Matbaacılık, ANKARA; 2009.
- Milli Eğitim Bakanlığı "Programlamaya Giriş ve Algoritmalar Ders Notları", 2007
- C Programlama Dili, Şerafettin ARIKAN
- Problem Solving and Program Design in C, Hanly, Koffman
- <http://www.AlgoritmaveProgramlama.com>



Algoritma ve Programlama

İYİ ÇALIŞMALAR...