

ALGORİTMA VE PROGRAMLAMA II

HAFTA#6

Genel Bakış...

2

- Veri Hiyerarşisi
- Dosyalara Giriş
- Dosyalar ve Akışlar
- Genel Dosya Komutları
- Dosyaya Yazma ve Okuma
 - fprintf ve fscanf fonksiyonlarının kullanımı

5. BÖLÜM

3

Dosya İşlemleri - Bölüm 1

Veri Hiyerarşisi

4

- Bir bilgisayardaki **en küçük veri parçası** **0** ya da **1** değerini alabilir.
- Bunun sebebi, iki kararlı durum içeren elektronik cihazları üretmenin **basit** ve **ekonomik** olmasıdır.
- Böyle veri parçalarına bit (**ikili basamak** anlamına gelen **binary digit** teriminin kısaltmasıdır, **basamak iki değerden birini alabilir**) denir.
- Bilgisayar devreleri, bir bitin değerini anlamak, *bite değer yerleştirmek* ve **bit değerlerini tersine çevirmek** (0 ise 1'e, 1 ise 0'a) gibi basit bit işlemlerini gerçekleştirirler.

Veri Hiyerarşisi (devam...)

5

- Programcılar için bitler biçimindeki düşük seviyeli verilerle çalışmak oldukça **zahmetlidir**.
- Bunun yerine, programcılar
 - **rakamlar** (yani 0, 1, 2, 3, 4, 5, 6, 7, 8, 9),
 - **harfler** (yani A-Z, a-z) ve
 - **özel semboller** (örneğin \$, @, %, &, *, ", :, ? ve diğerleri) formundaki **verilerle çalışmayı tercih ederler**.
- Rakamlar, harfler ve özel semboller **karakterler** olarak adlandırılırlar.

Veri Hiyerarşisi (devam...)

6

- Bilgisayarlar *yalnızca 1 ve 0'ları işleyebildiğinden*, **her karakter**, 1 ve 0'ların **değişik biçimde dizilişleriyle temsil edilir**.
- Yazılımcılar programlarını ve veri parçalarını **karakterlerle yaratır** ve bilgisayarlar da bu karakterleri, *bitlerin dizilişleri biçiminde yönetir* ve **işlerler**.

Dosyalara Giriş

7

- Değişkenler ve diziler içinde depolanan veriler **bellekte tutulurlar** ve **geçicidirler**.
- Bu türde veriler program sonlandığında **bellekten kaybolurlar**.
- **Dosyalar** *büyük miktarda veriyi* **kalıcı** olarak tutmak için kullanılır.
- Bilgisayarlar dosyaları *ikincil depolama cihazlarında*, özellikle de **disk depolama** cihazlarında tutarlar.

Dosya ve Akışlar (Stream)

8

- C, her dosyayı basit olarak **bitlerin art arda geldiği bir akış** olarak görür.
- Her dosya ya *dosya sonu belirteci* (***end-of-file***) ya da sistemde yönetici veri yapısı tarafından belirlenmiş **özel bir byte sayısı** ile sonlanır.
- Bir dosya *açıldığında*, dosya ile ilgili bir akış ilişkilendirilir. Program çalışmaya başladığında, **üç dosya** ve bu dosyalarla ilişkili akışlar;
 - **standart giriş** (*standart input*)
 - **standart çıkış** (*standart output*)
 - **standart hata** (*standart error*)

Dosya ve Akışlar (devam...)

9

- Akışlar, **dosyalar** ile **program** arasında haberleşme kanalları oluşturur. Örneğin, standart giriş akışı programın klavyeden **veri okumasını** ve standart çıkış akışı programın **ekrana veri yazdırmasını** sağlar.

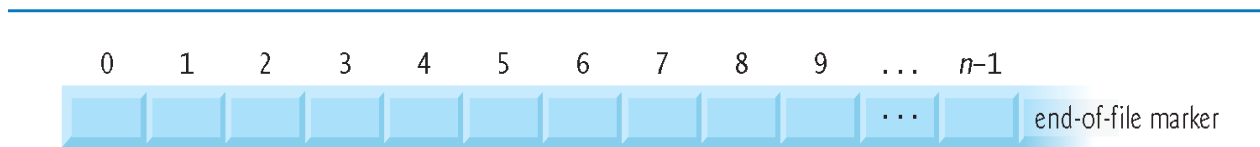


Fig. 11.1 | C's view of a file of n bytes.

Dosya ve Akışlar (devam...)

10

Standart kütüphane, dosyalardan **okuma yapmak** ve dosyalara **veri yazmak** için bir çok fonksiyon sunmaktadır.

- **fgetc** fonksiyonu, **getchar** gibi, dosyadan bir **karakter** okur.
- **fputc** fonksiyonu, **putchar** gibi dosyadan okunan karakteri **output** olarak ortaya koyar.
- **fgets** ve **fputs** fonksiyonları **dosyadan bir satır okumak** ya da **dosyaya bir satır yazdırmak** için kullanılır.

Genel Dosya Komutları

11

- Bir dosyaya okuma ve yazma yapmak için ilk işlem dosyayı açmaktır.
 - Dosya açmak için **fopen()**
 - Kapatmak için **fclose()**
 - Bu fonksiyonlar **stdio.h** kütüphanesi içerisinde tanımlanmıştır.

Genel Dosya Komutları (devam...)

12

- Genel olarak **dosya açma ve kapatma** adımları şu şekildedir.

```
FILE *pDosya; /* dosya işaretçisi */  
pDosya = fopen(const char dosya_adı,  
               const char mod);
```

...

...

```
fclose(pDosya);
```

Genel Dosya Komutları (devam...)

13

```
FILE *pDosya; /* dosya işaretçisi*/
```

- ifadesi, **pDosya**'nın **FILE** yapısını gösteren bir **işaretçi** olduğunu belirtmektedir.
- C programı, her dosyayı ayrı bir **FILE** yapısıyla yönetir.
- Yazılımcı dosyaları kullanabilmek için **FILE** yapısının özelliklerini bilmelidir.

Genel Dosya Komutları (devam...)

14

- Dosya açma işlemi için kullanılan **fopen()** fonksiyonundaki **mod** değişkeni için değişkenler ve tanımları aşağıdaki tabloda belirtilmiştir.

Açılış Modu	İşlem Türü
r	Salt okunur (read only). Dosyanın açılabilmesi için önceden oluşturulmuş olması gerekir. Bu modda açılmış olan bir dosyaya yazma yapılamaz.
w	Yalnızca yazma (write only). Dosya diskte kayıtlı olsun veya olmasın dosya yeniden oluşturulur. Bu modda açılmış olan bir dosyadan okuma yapılamaz.
a	Ekleme (append). Kayıtlı bir dosyanın sonuna veri eklemek için açılır. Bu modda açılmış olan bir dosyadan okuma yapılamaz.
r+	Okuma ve yazma. Bu modda açılmış olan bir dosyanın daha önce varolması gerekir.
w+	Okuma ve yazma. Bu modda açılmış olan bir dosya var olsun veya olmasın dosya yeniden oluşturulur.
a+	Okuma ve yazma. Kayıtlı bir dosyanın sonuna veri eklemek için açılır.

Genel Dosya Komutları (devam...)

15

- Bir dosyaya erişmek ve üzerinde işlem yapabilmek için ise o *dosyanın açılıp açılmadığını* **test etmek** gerekir.

```
int main()
{
    FILE *pDosya;

    pDosya = fopen("dosya.txt", "r");

    if(pDosya == NULL)
    {
        printf("dosya.txt dosyasi acilmadi. !\n");
        exit(1);
    }

    /* Dosya islemleri gerceklestiriliyor */
    /* ----- */

    fclose(pDosya);
    return 0;
}
```

fprintf() ve fscanf()

16

- fprintf() ve fscanf() fonksiyonları
 - dosyaya veri yazma ve
 - dosyadan veri okuma işlemlerinde kullanılırlar.
- Veriler üzerinde değişim yaparak çalışırlar.

fprintf() ve fscanf() (devam...)

17

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

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

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

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

Örnek 1: fprintf() ile Dosyaya Veri Yazma

18

- Okuma ve yazma modunda “**hangiyildayiz.txt**” isimli bir dosya yaratınız.
- Dosyanın içerisine **fprintf()** kullanarak:
 - “**Bu sene 2015 yilindayiz**” yazdırınız.
 - “**İki string**” ve “**bir integer**” kullanılacaktır.
- İşiniz bittikten sonra dosyayı **fclose()** ile kapatmayı unutmayınız.

Örnek 1: fprintf() ile Dosyaya Veri Yazma

19

```
int main()
{
    FILE *fp;

    fp = fopen("hangiylidayiz.txt", "w+");
    if(fp == NULL )
    {
        printf("hangiylidayiz.txt dosyasi acilmadi. !\n");
        exit(1);
    }

    fprintf(fp, "%s %d %s", "Bu sene", 2015, "yilindayiz");
    fclose(fp);

    return 0;
}
```

Örnek 2: fscanf() ile Dosyadan Veri Okuma

20

- Daha önce oluşturduğumuz “**hangiyildayiz.txt**” isimli bir dosyayı **fscanf()** kullanarak okuyalım.
 - **Yıl bilgisini bularak ekrana yazdıralım.**
- İşiniz bittikten sonra dosyayı **fclose()** ile kapatmayı unutmayınız.
- **Dikkat:** “Bu sene” ifadesinde arada boşluk var. **%s** ile okurken boşluklar alınmaz.

Örnek 2: fscanf() ile Dosyadan Veri Okuma

21

```
int main()
{
    FILE *fp;

    fp = fopen("hangiylidayiz.txt", "r");

    if (fp == NULL)
    {
        printf("Dosya okunamadi !!! \n");
        exit(1);
    }

    int yil;
    char s1[10];
    char s2[10];

    fscanf(fp, "%s %d %s", s1, &yil, s2);

    printf("Yil: %d", yil);
    return 0;
}
```

“Bu sene” ifadesinde boşluğu silin.

Örnek 3: Dosyaya Veri Yazma ve Dosyadan Veri Okuma

22

- 2 tane öğrencinin aşağıdaki bilgileri klavyeden okunacak ve daha sonra “**ogrenci.txt**” isim dosyaya yazdırılacaktır.
 - No: int
 - Ad: char[50];
 - Not: int
- Klavyeden okuma ve dosyaya yazma işlemleri için aşağıdaki fonksiyon kullanılacaktır.
 - **void** DosyaYaz()
- Daha sonra **dosya okuma işlemi** yapılarak, okunan bilgiler ekrana yazdırılacaktır.
 - **void** DosyaOku()

Örnek 3: Dosyaya Veri Yazma ve Dosyadan Veri Okuma

23

```
OGRENCI BILGILERINI GIRINIZ:
-----
1. OGRENCI NO      : 1
1. OGRENCI ADI     : Deniz
1. OGRENCI NOTU    : 100

2. OGRENCI NO      : 2
2. OGRENCI ADI     : Ada
2. OGRENCI NOTU    : 99

OGRENCI BILGILERI KAYDEDILDI

OGRENCI BILGILERI DOSYADAN OKUNUYOR

OGRENCI BILGILERI
-----
NO=1
AD=Deniz
NOT=100

NO=2
AD=Ada
NOT=99
```

KAYNAKLAR

24

- 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...