

# **Bilgisayar Bilimlerine Giriş-II**

**-6-**

**BIL 1002**

**Dokuz Eylül Üniversitesi, Fen Fakültesi,  
Bilgisayar Bilimleri Bölümü**

# Dosyalar (FILE)

- ▶ Bir veriyi *dosyaya yazmak* aslında onu sabit disk üzerine kaydetmektir.
- ▶ Program sonlandığında kaybolması istenmeyen verilerin disk üzerinde saklanması gerekir.
- ▶ Ayrıca, bazen RAM boyutu yetersiz kaldığında, her ne kadar disk üzerinde okuma/yazma RAM ve diğer belleklere göre çok daha yavaş olsa da, program için gerekli değerler, değişkenler yerine dosyalar üzerinde saklanıp gerektiğinde kullanılabilir.

# FILE İşaretçisi

► **FILE** yapısı **stdio.h** da tanımlanmış dosya kontrol yapısıdır ve *dosya işaretçisi (file pointer)* olarak adlandırılır.

► Bir dosya işaretçisi aşağıdaki gibi tanımlanır

**FILE \*dosya\_işaretçisi;**

Burada *dosya\_işaretçisi* bir işaretçi değişken gibidir.

# Dosya Türleri

- ▶ Bir dosya içerdiği veriye göre;
  - ▶ *metin dosyası* ya da
  - ▶ *ikili dosya* türündedir.
- ▶ Programda kullandığımız fonksiyon dosyaya ne tipte veri yazılacağını belirleyecek ve okuma da buna göre yapılacaktır.

# Dosyayı Oluşturmak (**fopen**)

- ▶ Bir dosyaya erişmek için önce onu açmamız gerek. Bunun için:  
*dosya\_isaretcisi* = **fopen**(*dosya\_adi*, *mod*)
- ▶ *dosya\_adi* açılacak dosyanın sistemde kayıtlı isminin yazıldığı bir katar (string) olmalıdır. Eğer dosya programın .exe dosyası ile aynı dizinde değilse, tam yolu yazılmalıdır.
- ▶ Fonksiyonun döndürdüğü *dosya\_isaretcisi* diğer dosya fonksiyonlarında kullanılmak üzere bir işaretçi değişkende saklanmalıdır. Daha sonra **fopen()** ile açılan dosya üzerinde *okuma/yazma* için yine aynı *dosya\_isaretcisi* kullanılacaktır.

# Modlar

*mod*, dosyanın ne amaçla açılacağını belirtir.

Mod	Tanım
"r"	Sadece okuma için açar. İmleç dosya başındadır.
"w"	Sadece yazma için açar. Aynı isimde bir dosya varsa, içeriği kaybolur; yoksa yeni bir dosya oluşturulur. İmleç dosya başındadır.
"a"	Ekleme yapmak için açar. Aynı isimde bir dosya varsa, içeriği korunur; yoksa yeni bir dosya oluşturulur. İmleç dosya sonundadır.
"r+"	Okuma ve yazma için açar. İmleç dosya başındadır.
"w+"	Yazma modu ("w") ile aynı özelliklere sahiptir. Okuma da yapabilir.
"a+"	Ekleme modu ("a") ile aynı özelliklere sahiptir. Okuma da yapabilir.

# Dosyayı Oluşturmak

- ▶ Eğer dosyaya erişilemediyse, fonksiyon **NULL** değerini döndürür. Genellikle dosya adının yanlış yazılması veya dosyanın zaten açık olması bu duruma yol açar.
- ▶ Programda mutlaka dosyanın açılıp açılmadığı kontrol edilmelidir.

```
FILE *f;  
f = fopen("deneme.dat", "r");  
if (f == NULL) {  
    printf("Dosya Acilamadi!\n");  
    return -1;    // rastgele verilmiş hata değeri  
}
```

# Dosya Kapatma (**fclose**)

- ▶ Dosyayla ilgili işlem bitirildiğinde dosya kapanmalıdır. Genelde sistemler, program tarafından açılan dosyayı programın sonlanmasıyla beraber kapatır. Ancak en doğrusu programcının bu dosyayı kendi kontrolünde kapatmasıdır.
- ▶ Ayrıca, daha önce açılmış bir dosyayı yeniden farklı bir modda açmak için önce o dosyayı kapatmak gerekir.
- ▶ **Dosya kapatma fonksiyonu:**

***fclose***(*dosya\_isaretcisi*)

Dosya hatasız kapandıysa 0 döndürür.



# Dosya Okuma ve Yazma

- ▶ Bir dosyaya veri kaydetmek için çeşitli yöntemlerden biri seçilebilir. Veri kaydetmek için C dilinde birçok fonksiyon bulunmaktadır. Bu fonksiyonlardan bazıları aşağıdaki gibi sınıflandırılabilir.
  - ▶ Dosyaya bir **karakterin** kaydedilmesi/okunması
  - ▶ Dosyaya bir **karakter dizisinin** kaydedilmesi/okunması
  - ▶ Dosyaya **blok bilgi** kaydedilmesi/okunması

# Dosyaya Bir Karakterin Kaydedilmesi/Okunması

- Bir metin dosyası üzerinde karakter düzeyinde **yazma** için:

*fputc(karakter, dosya\_isaretcisi)*

- Bir metin dosyası üzerinde karakter düzeyinde **okuma** için:

*karakter = fgetc(dosya\_isaretcisi)*

fonksiyonları kullanılır.

# Örnek:

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

```
int main()
```

```
{
```

```
    FILE *fp;
```

```
    int ch;
```

```
    char c;
```

```
    fp = fopen("test1.txt", "w");
```

```
    if(fp==NULL){
```

```
        printf("Dosya yazdirma icin acilamadi");
```

```
        return -1;
```

```
    }else
```

```
        for( ch = 'a' ; ch <= 'z'; ch++ )
```

```
        {
```

```
            fputc(ch, fp);
```

```
        }
```

```
        fclose(fp);
```

```
fp=fopen("test1.txt","r");
```

```
printf("Dosyadaki Bilgiler");
```

```
while(!feof(fp)){
```

```
    c=fgetc(fp);
```

```
    printf("%c\n",c);
```

```
}
```

```
return(0);
```

```
}
```

# Dosyaya Bir Karakter Dizisinin Kaydedilmesi/Okunması

- Bir metin dosyası üzerinde katar (string) düzeyinde **yazma** için:

*fputs(katar, dosya\_isaretcisi)*

- Bir metin dosyası üzerinde katar (string) düzeyinde **okuma** için:

*fgets(katar, adet, dosya\_isaretcisi)*

- *katar* parametresi yazılacak/okunacak karakter katarı değişkenidir.
- Okuma fonksiyonunda *adet* parametresi kaç karakter okunacağını belirtir.

# Dosyaya Bir Karakter Dizisinin Kaydedilmesi/Okunması

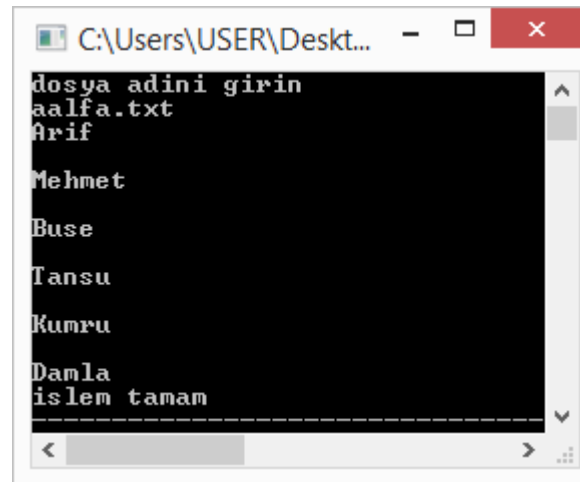
- ▶ Katarı dosyaya yazan fonksiyonun, fputs()’un 2 parametresi vardır. Biri yazılacak katarın başlangıç adresi; diğeri dosya işaretçisidir. Aşağıda klavyeden katar okunmuş ve dosyaya yazılmıştır. Yazma işlemi, katar değişkene ayrılan yer kadar değil (örnekte 256), satır sonu karakteriyle(NULL) karşılaşılan kadar yapılır.

- ▶ FILE \*di;  
char satir[256];  
-----  
gets(satir);  
-----  
fputs(satir,di);  
-----

```
► FILE *di;  
   int n;  
   char satir[256];  
   .....  
   while(fgets(satir,n,di)!=NULL)  
       puts(satir);
```

Burada en fazla (n-1) karakter okunup, satir adlı katar değişkene yerleştirilir. Dosya sonuna gelinip gelinmediği sınanmasını ayrıca yapmak gereksizdir. Çünkü dosya sonuna gelinmişse, okuma fonksiyonu NULL göndermektedir.

```
► int main() {  
    char sss[81],string[256],c;  
    FILE *di;  
    printf("dosya adini girin\n");  
    gets(sss);  
    di=fopen(sss,"r");  
    if(di==NULL){  
        printf("Dosya acilamadi\n");  
        exit(0);  
    }  
    while(fgets(string,256,di)!=NULL){  
        // printf("%s ",string);  
        puts(string);  
    }  
    printf("islem tamam");  
    fclose(di);  
    return 0;  
}
```



```
C:\Users\USER\Desktop...  
dosya adini girin  
aalfa.txt  
Arif  
Mehmet  
Buse  
Tansu  
Kumru  
Damla  
islem tamam
```

# Dosyaya Blok Bilgi

## Kaydedilmesi/Okunması

- Bir metin dosyası üzerinde formatlı **yazma** için:

*fprintf*(*dosya\_isaretcisi*, *formatlı\_erisim*)

- Bir metin dosyasından formatlı **okuma** için:

*fscanf*(*dosya\_isaretcisi*, *formatlı\_erisim*)

fonksiyonları kullanılır.

- *formatlı\_erisim* parametresi, standart **girdi/çıkı**tı fonksiyonları olan **scanf** ve **printf** gibi, format belirteçleri ve ilgili değişkenlerden oluşan parametrelerin fonksiyona alınacağını belirtir.

- İmleci, dosyanın başına geri döndürmek için:

*rewind*(*dosya\_isaretcisi*)

fonksiyonu kullanılır.



# Dosyaya Blok Bilgi

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

```
int main ()
```

```
{
```

```
    char str [80];
```

```
    float f;
```

```
    FILE * pFile;
```

```
    pFile = fopen ("myfile.txt","w+");
```

```
    fprintf (pFile, "%f %s", 3.1416, "PI");
```

```
    rewind (pFile);
```

```
    fscanf (pFile, "%f", &f);
```

```
    fscanf (pFile, "%s", str);
```

```
    fclose (pFile);
```

```
    printf ("Okunan degerler: %f ve %s \n",f,str);
```

```
    return 0;
```

```
}
```

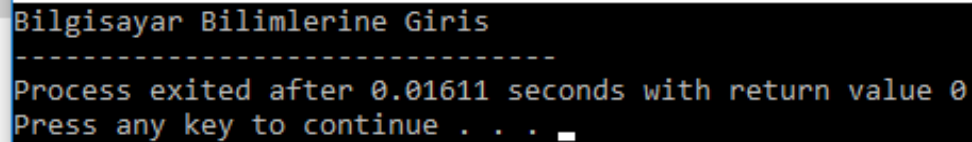
# **feof()** Fonksiyonu

- ▶ *feof()* fonksiyonu dosyadaki bilgilerin tümünün okunup okunmadığını gösterir. Bir dosya açıldığında bilgiler sırayla okunur. Dosya işaretcisi her okumada yeni okuyacağı adresi alır. Herhangi bir kontrol yapılmaz ise disk üzerinde bu işlem kayıt ünitesi üzerindeki adreslerin sonuna kadar sürer. Dosyanın sonuna gelinip gelinmediği bu fonksiyon ile kontrol edilebilir.
- ▶ Fonksiyonun kullanımını aşağıdaki

*int feof(dosya\_işaretcisi);*

# feof() Fonksiyonu

```
#include <stdio.h>
int main ()
{
    FILE *fp;
    int c
    fp = fopen("file.txt","r");
    if(fp == NULL)
    {
        perror("Error in opening file");
        return(-1);
    }
    while(1)
    {
        c = fgetc(fp);
        if( feof(fp) )
        {
            break ;
        }
        printf("%c", c);
    }
    fclose(fp);
    return(0);
}
```

A terminal window with a black background and green text. The text shows the program's output: the title 'Bilgisayar Bilimlerine Giris', a separator line of dashes, the message 'Process exited after 0.01611 seconds with return value 0', and a prompt 'Press any key to continue . . . ' followed by a cursor.

```
Bilgisayar Bilimlerine Giris
-----
Process exited after 0.01611 seconds with return value 0
Press any key to continue . . . _
```

ÖRNEKLER

**Soru 1:** Bir metin dosyasının içeriğini karakter düzeyinde okuyup ekrana yazdıran bir C programı yazın.

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

```
int main() {
```

```
FILE *f;
```

```
// dosya programın derlendiği dizinde bulunmalı
```

```
    f = fopen("deneme.txt", "r");
```

```
    if (f == NULL) {
```

```
        printf("Dosya Acilamadi!\n");
```

```
        getchar();
```

```
        return -1;
```

```
    }
```

```
    while (!feof(f))
```

```
        printf("%c", getc(f));
```

```
    fclose(f);
```

```
    getchar();
```

```
    return 0;
```

```
}
```

**Soru 2:** 1-100 arasında rastgele sayıda olmak üzere İngiliz alfabesinden küçük harfler üreterek bir metin dosyasına yazıp kaydettikten sonra bu dosyanın içeriğini okuyup ekrana yazdıran bir C programı yazın.

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
int main() {
    FILE *f;
    srand(time(NULL));
    int i;
    int n = 1 + rand()%100;
    // dosya programın derlendiği dizinde yaratılır
    if ((f = fopen("harfler.txt", "w")) == NULL) {
        printf("Dosya Acilamadi!\n");
        getchar();
        return 0;
    }
    for (i = 0; i < n; i++)
        putc('a' + rand()%26, f);
    fclose(f);

    if ((f = fopen("harfler.txt", "r")) == NULL) {
        printf("Dosya Acilamadi!\n");
        getchar();
        return 0;
    }
    while ((i = getc(f)) != EOF)
        putchar(i);
    fclose(f);
    getchar();
    return 0;
}
```

**Soru 3:** Bir metin dosyasına, kullanıcı girişi yaptığı sürece (belirsiz sayıda) numarasını ve 4 üzerinden reel sayı puanını alacağı öğrencilerin bilgilerini formatlı erişim yoluyla yazıp okuyarak en sonda ekrana toplu halde yazdıran bir C programı yazın.

```
#include <stdio.h>
#include <conio.h>
#include <cstdlib>
int main() {
    FILE *f;
    char ad[30];
    int no, i, k = 0;
    if ((f = fopen("ogrenciler.txt", "w+")) == NULL) {
        printf("Dosya Acilamadi!\n"); getchar();
        return 0;
    }
    do {
        printf("\n\nOgrenci adi:\t"); scanf("%s", ad);
        printf("Ogrenci no:\t"); scanf(" %d",
            &no);
        fprintf(f, "%s %d", ad, no);
        printf("Ogrenci kaydina devam etmek icin d
            tusuna basin\t");
        k++;
    } while (getche() == 'd');

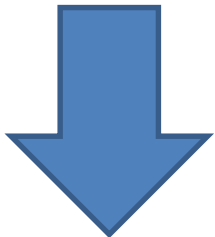
    rewind(f);
    // ekranı temizlemek için sistem çağrısı
    system("cls");
    for (i = 1; i <= k; i++) {
        fscanf(f, "%s %d", ad, &no);
        printf("%d. Ogr:\t%s\tNo: %d\n", i, ad, no);
    }
    return 0;
}
```

**Soru 4:** Klavyeden öğrencinin **ad**, **V** ve **F** notu bilgilerini alan, daha sonra **gn=v\*0.4+f\*0.6** şeklinde geçme notu hesaplatan, “öğrenci.txt” isimli dosyaya ad ve gn kaydeden, sınıfın ortalamasını hesaplatıp ekranda yazdıran, ortalamadan küçük olan uygun isimleri ve gn notlarını “kalanlar.txt” isimli dosyaya, ortalamadan büyük olan uygun isimleri ve gn notlarını “gecenler.txt” isimli dosyaya kaydeden C programı yazınız.

```
#include<stdio.h>
#include<conio.h>
int main()
{
    FILE *d1,*d2,*d3;
    int v,f,k=0;
    float gn,s=0,ort;
    char ad[10],cev;
    d1=fopen("ogrenci.txt","w");

    do{
        printf("\nOğrenci ad, V ve F notu giriniz:");
        scanf("%s %d %d",ad,&v,&f);
        k++;
        gn=v*0.4+f*0.6;
        s+=gn;
        printf("\nGecme notu:%f",gn);
        fprintf(d1,"%s %f\n",ad,gn);
        printf("\nBaska ogrenci...");
        cev=getch();
    }while(cev=='E' || cev=='e');

    fclose(d1);
    ort=(float)s/k;
    printf("\nOrtalama=%f",ort);
```





```
d1=fopen("ogrenci.txt","r");
d2=fopen("gecenler.txt","w");
d3=fopen("kalanlar.txt","w");
while(!feof(d1))
{
    fscanf(d1,"%s %f\n",ad,&gn);
    if(gn>=ort) {
        fprintf(d2,"%s %f\n",ad,gn);
    }else{
        fprintf(d3,"%s %f\n",ad,gn);
    }
}

fclose(d2);
fclose(d3);
fclose(d1);
return 0;
}
```

SON