

# C

# Dosyalama

Dr. Öğr. Üyesi M. Ozan AKI

# Dosyalama

Dosyalar, disk üzerinde verilerin kalıcı olarak saklanması ve başka bir ortama kopyalanarak taşınabilmesini sağlayan yapılardır.

# Dosyalama

C Dili ve Birçok Programlama Dilinde Dosyalar **Metin (Text)** ve **İkili (Binary)** olarak iki tür olarak ele alınır ve kullanılır.

**Metin (Text) Dosyalar:** Notepad ile açılıp okunabilen, karakter dizilerinden oluşan satırlar halinde düzenlenmiş ve her satır sonunda CR LF (Enter) karakteri bulunan dosyalardır.

**İkili (Binary) Dosyalar:** Ancak formatına uygun bir şekilde yazılıp okunabilen dosyalardır. Dosya formatı, dosyanın hangi noktasında hangi verinin olduğunu belirleyen bildirgelerdir.

# Dosyalamada Kullanılan Fonksiyonlar

- FILE\*** : Dosya işaretçilerini saklamak için pointer tipinde değişkendir.
- fopen()** : Bir dosya açar
- fclose()** : Açık bir dosyayı kapatır
- fputc()** : Dosyaya bir karakter yazar
- fgetc()** : Dosyadan bir karakter okur
- fputs()** : Dosyaya string yazar
- fgets()** : Dosyadan string okur
- fseek()** : Dosya içinde istenen konuma gider
- ftell()** : Dosya içindeki geçerli konumu döndürür
- fprintf()** : Dosyaya formatlı bilgi yazar
- fscanf()** : Dosyadan formatlı bilgi okur
- feof()** : Dosya sonuna gelindiğinde true döndürür
- ferror()** : hata meydana gelmişse true döndürür
- rewind()** : Dosya konum göstergesini dosya başına alır
- remove()** : Dosyayı siler
- fflush()** : Dosya tampon belleğindeki verileri dosyaya yazarak boşaltır

# fopen()

**FILE\* fopen(char \*filename, char\* mode);**

**filename:** Yolu ile beraber Dosya adı

**mode** : Dosya Kipini belirtir

- **r** : Metin dosyasını okuma için açar
- **w** : Metin dosyasını yazma için oluşturur
- **a** : Metin dosyasının sonuna ekler
- **rb** : İkili dosyayı okuma için açar
- **wb** : İkili dosyayı yazma için oluşturur
- **ab** : İkili dosyanın sonuna ekler
- **r+** : Metin dosyasını okuma/yazma için açar
- **w+** : Metin dosyasını okuma/yazma için oluşturur
- **a+** : Metin dosyasını okuma/yazma için oluşturur, sonuna ekler
- **r+b** : İkili dosyayı okuma/yazma için açar
- **w+b** : İkili dosyayı okuma/yazma için oluşturur
- **a+b** : İkili dosyası okuma/yazma için oluşturur, sonuna ekler

# fclose()

**int fclose(FILE\* dosya);**

Parametre ile verilen dosyayı kapatır. Eğer dosya kapatma işlemi başarılı olduysa 0 (sıfır) döndürür. Başarısız olduysa **EOF** döndürür.

Örneğin USB disk üzerinde açılan bir dosya, USB diskin zamansız çıkarılması sonucunda dosya düzgün kapatılamaz. Üstelik tampon bellekte bulunan bilgiler dosyaya henüz yazılmamış olabilir.

# fopen(), fclose()

```
FILE* dosya;
```

```
dosya = fopen("C:\deneme.txt", "w+");  
if(dosya == NULL)  
{  
    printf("Hata: Dosya oluşturulamadı!\n");  
    return;  
}
```

```
... // Dosya ile ilgili işlemler
```

```
fclose(dosya);
```

**fputc() fgetc()**

**int fputc(int karakter, FILE\* dosya);**

parametre ile verilen dosyanın geçerli konumuna bir karakter yazar. Başarılı olursa **0**, Yazamaz ise **EOF** döndürür.

**int fgetc(FILE\* dosya);**

parametre ile verilen dosyanın geçerli konumundan bir karakter okur ve ASCII kodunu döndürür. Eğer okunamaz ise **EOF** döndürür



## **fprintf() fscanf()**

**void fprintf(FILE\* dosya, char\* format, args);**

Tıpkı konsol ekranında kullanılan printf komutu gibi çalışır. Sadece formatlı verileri ekrana değil, dosyaya yazar.

**int fscanf(FILE\* dosya, char\* format, args);**

Tıpkı konsol ekranında kullanılan scanf gibi çalışır. Formatlı verileri ekrandan değil, dosyadan okur.

# rewind(), feof()

**void rewind(FILE\* dosya);**

Parametre ile verilen dosyanın imleç konumunu dosya başına alır.

**int feof(FILE\* dosya);**

parametre ile verilen dosyanın sonuna gelindiğinde TRUE (sıfırdan farklı) değer döndürür, aksi halde 0 (sıfır) döndürür.

# İkili (Binary) Dosyalar

Binary dosyalarda, text dosyalarda olduğu gibi herahngi satır, satır sonu, okunabilir ASCII karakterler sözkonusu değildir.

Okuma ve Yazma işlemleri byte olarak yapılır. Dolayısıyla örneğin bir integer değer yazmak için dosyaya 4 byte yazılmalı, okurkende 4 byte okunmalıdır.

# fread(), fwrite()

**size\_t fread(void\* bellek, size\_t boyut, size\_t, sayac, FILE\* dosya);**

Parametre olarak, okunacak verinin saklanacağı bellek, okunacak veri tipi boyutu, veri adedi ve dosya işaretçisi verilir. Okunan veri adedini döndürür.

**size\_t fwrite(void\* bellek, size\_t boyut, size\_t, sayac, FILE\* dosya);**

Parametre olarak, yazılacak verinin tutulduğu bellek, yazılacak veri tipi boyutu, veri adedi ve dosya işaretçisi verilir. Yazılan veri adedini döndürür.