



5. HAFTA

BLM112

PROGRAMLAMA DİLLERİ II

Yrd. Doç. Dr. Ümit ATİLA

umitatila@karabuk.edu.tr

KBUZEM

Karabük Üniversitesi

Uzaktan Eğitim Araştırma ve Uygulama Merkezi

1. Dosyalama

Yazdığımız programlar geçici bellekte çalıştırılmakta ve program kapatıldığında bütün veriler silinmektedir. Saklanması gereken veriler kalıcı hafızaya yazılmalıdır. Bunun için dosyalama işlemleri kullanılır. Dosyalarda veriler belirli bir düzen içerisinde saklanır. Her dosyanın kendine özgü bir yapısı vardır. Microsoft tabanlı işletim sistemleri dosyanın tipini uzantısına göre sınıflandırmaktadır. Örneğin .doc uzantısı kelime işlemci(word), .xls uzantısı çalışma sayfası(excel), .cpp C++ dosyası olarak sınıflandırılmaktadır.

C dilinde farklı dosyalama tipleri bulunmaktadır. Bunlar:

1. Sıralı Erişimli (Text) dosyalar
2. Rastgele Erişimli (Random) dosyalar
3. İkili (Binary) Dosyalar

1.1. Dosya Açma Fonksiyonları

fopen : dosya açmak için kullanılır.

Tanımlama biçimi:

```
FILE *fopen (char *Dosya_Adi, char *Parametre);
```

FILE	: Dosyalar üzerindeki işlemleri için kullanılan yapı
fopen	: Dosya açma fonksiyonu
Dosya_Adi	: Açılacak dosya adı
Parametre	: Dosyanın hangi modda/amaçla açılacağını belirleyen karakterler

fopen fonksiyonu ile kullanılan parametreler Çizelge 1'de gösterilmektedir.

Çizelge 1:

r	Mevcut dosyanın yalnız okunmak üzere açılmasında kullanılır.
w	Dosyanın yazmak için açıldığını belirtmek için kullanılır. Eğer daha önce aynı isimde bir dosya varsa, eski dosya silinerek yeni dosya baştan oluşturulur.
a	Açılmış olan bir dosyanın sonuna bilgi ilave etmek için kullanılır. Eğer dosya yoksa, oluşturulur.
r+	Mevcut dosya üzerinde hem okuma, hem yazma yapmak için kullanılır.
w+	Dosya okumak ve yazmak için yeniden açar. Dosya daha önceden varsa, mevcut bilgiler silinecektir. Bu parametre ile açılan dosyalara kayıt yapılabilir ve yapılan kayıtlar okunabilir.
a+	Mevcut dosyayı okumak ve yeni bilgi ilave etmek için kullanılır. Eski bilgiler silinmez.

Örnek:

```
#include "stdio.h"
#include "stdlib.h"

int main()
{
    FILE *dosya, *dosya1, *dosya2;
    dosya = fopen("deneme.dat", "r+"); //w+'yı deneyiniz
    dosya1 = fopen("rehber.dat", "w+");
    dosya2 = fopen("adres.dat", "a+");

    if( (dosya == NULL) || (dosya1 == NULL) || (dosya2 == NULL))
    {
        puts("Dosyalardan biri acilmiyor...");
        getch();
        exit(1);
    }
    else printf("dosyalar acildi");

    return 0;
}
```

Program ilk çalıştırıldığında deneme.dat isimli dosya yoksa hata verecektir. Dosyalar ana programla aynı dizinde olmalıdır. Aksi halde dosya yolu belirtilmelidir.

Örneğin dosyanın yolu: `C:\WINDOWS\DESKTOP\deneme.dat` ise dosya açılırken

```
dosya = fopen("C:\\WINDOWS\\DESKTOP\\deneme.dat", "w");
```

şeklinde açık yol bildirilmelidir.

1.2. Dosya Kapatma Fonksiyonları

C dilinde iki tane dosya kapatma fonksiyonu vardır. Bunlar `fclose` ve `fcloseall` fonksiyonlarıdır. Bir dosyanın kapatılabilmesi için öncelikle dosyanın açılmış olması gerekir. Dosyanın kapatılması, bilgilerin diske kaydedilmesi ve dosya ile ilişkinin kesilmesinden sonra olur.

```
int fclose (FILE *Dosya);
```

Örnek:

```
#include "stdio.h"
#include "stdlib.h"
int main()
{
    FILE *dosya;
    dosya = fopen("deneme.dat", "r+");
    fclose(dosya);
    return 0;
}
```

```
int fcloseall (FILE *Dosya); //Yalnız TurboC'de
```

Örnek:

```
#include "stdio.h"
#include "stdlib.h"
int x;
FILE *dosya, *dosya1, *dosya2;
int main()
{
    dosya = fopen("deneme.dat", "r+");
    dosya1 = fopen("rehber.dat", "w+");
    dosya2 = fopen("adres.dat", "a+");
    x = fcloseall();
    printf("Kapatılan dosya sayısı= %d\n", x);
    getch();
    return 0;
}
```

1.3. Dosyaya Yazma ve Dosyadan Okuma Fonksiyonları

Fonksiyon	Görevi
putc()	Dosyaya bir karakter yazar
getc()	Dosyadan bir karakter okur
fprintf()	Dosyaya formatlı veri yazar
fscanf()	Dosyadan formatlı veri okur
fputs()	Dosyaya katar yazar
fgets()	Dosyadan katar okur
fwrite()	Dosyaya dizi yazar
fread()	Dosyadan dizi okur

1.4. Dosya Silme Fonksiyonları

C dilinde dosya silmek için **remove** ve **unlink** fonksiyonları kullanılır. Kullanımı aşağıdaki gibidir.

```
int remove (char *Dosya_adi);  
int unlink(char *Dosya_adi);
```

Bu fonksiyonlar sorunsuz çalışırsa sıfır değeri geri döner.

1.5. Sıralı Erişimli (Text) Dosyalar

Sıralı erişimli dosyalar karakterlerden oluşur. Bu tip dosyalarda satır sonu "\n" özel karakteri ile biter. Ayrıca dosyanın sonuna gelindiğini de "EOF" özel karakteri bildirir.

Text dosyalarda kayıt okuma işlemi birinci satırdan başlayıp, en son satırda sona erer. Herhangi bir kayda ulaşmak için bütün kayıtların okunması gerekmektedir. Bu yüzden text dosyalara sıralı erişimli dosyalarda denir. Programın çalışmasının yavaş olması sebebi ile çok fazla tercih edilmezler.

Text dosyalarda sayısal ifadeler formatlı yazdırılmak zorundadır. Aksi halde sayısal değerler ile ilgili problem yaşanabilir. Text dosyalar ASCII kodlarını kullandığından kayıt sonlarının "\n" karakteriyle bittiği unutulmamalı ve gerektiğinde kullanılmalıdır.

Örnek: Öğrenci bilgilerini dosyaya kayıt eden program

```
#include <stdio.h>
#include <stdlib.h>
#include <conio.h>
int main()
{
    FILE *dosya; /* dosya göstericisi */
    typedef struct
    {
        char ad[10];
        int no, Not;
    } ogrenci;
    ogrenci kisi;
    dosya = fopen("sinif.txt", "w");
    if( dosya == NULL )
        puts("sinif.txt dosyasi acilmadi. !\n"), exit(1);
    printf("ogrencinin adi      : ");
    gets(kisi.ad);
    printf("ogrencinin numarası: ");
    scanf("%d",&kisi.no);
```

```

    printf("ogrencinin notu      : ");
    scanf("%d",&kisi.Not);
    fprintf(dosya,"%5d %10s
%3d\n",kisi.no,kisi.ad,kisi.Not); /* verileri
formatlı yaz! */
    fflush(stdin); //açılan dosyanın tampon
belleğindeki bilgileri diske aktarır.
    /* dosyayı kapat */
    fclose(dosya);
    puts("Bilgiler kaydedildi.\a");
    getch();
    return 0;
}

```

Örnek: Kaydedilmiş öğrenci bilgilerini dosyadan okuyup ekranda görüntüleyen program

```

#include <stdio.h>
#include <stdlib.h>
#include <conio.h>
int main()
{
    FILE *dosya;                /* dosya göstericisi */
    typedef struct
    {
        char  ad[10];
        int   no, Not;
    } ogrenci;
    ogrenci kisi;
    dosya = fopen("sinif.txt", "r");
    if( dosya == NULL )
        puts("sinif.txt dosyasi acilamadi. !\n"),
        exit(1);
    while(!feof(dosya))
    {
        fflush(stdout);
        fscanf(dosya,"%d %s %d\n",&kisi.no, &kisi.ad,
&kisi.Not);
        printf("\n");
        printf("%d %s %d\n",kisi.no, kisi.ad,
kisi.Not);
    }
}

```

```
fclose(dosya);  
getch();  
return 0;  
}
```