



BİLGİSAYAR PROGRAMLAMA 1

Ders Notu 13 – Dosya İşlemleri

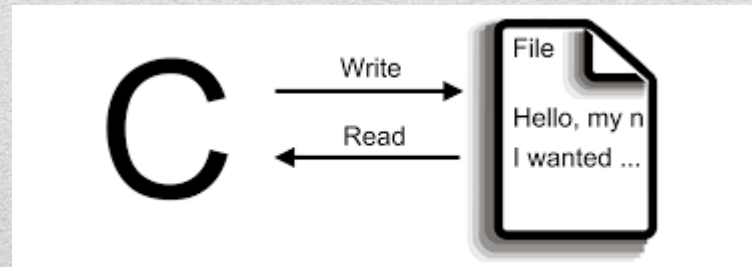
Konya Teknik Üniversitesi
Elektrik – Elektronik Mühendisliği Bölümü

27.05.2024

Konya

Dosya İşlemleri

- Değişkenlerden farklı olarak bir dosya, programın yürütülmesinden önce ve/veya sonra da var olmaktadır.
- Dosyalar sayesinde başka programlar da birbirleriyle haberleşebilirler, bunlar arasında veri aktarımı sağlanır.
- Dosya boyutlarınının sabit diskin izin verdiği ölçüde sınırsız genişlikte olması da bunları kullanmamız için önemli nedenlerdendir.



Dosya Türleri

- İşletim sistemleri dosya türlerinin **Metin Dosyası** ve **İkili(Binary) Dosyalar** olarak iki farklı biçimde saklanmasına izin verirler.
- Metin dosyaları sadece ASCII karakterlerinden oluşabilir.
- Bundan dolayı ASCII dosyaları şeklinde de adlandırılabilirler.
- Her ASCII karakteri 1 bayt olduğundan metin dosyasında kaç karakter varsa boyutu da o kadar bayttır.
- Metin dosyaları sadece ileri yönlüdür, mesela 3 satırdan oluşan bir metin dosyasının 2. satırına metin ekleyemezsiniz, en sona ekleyebilirsiniz.
- İkili dosyalar baytlar halinde okunur ve yazılırlar. 8 bitlik 1 ve 0'lar.
 - Özel editörler ile okunur ve yazılırlar.
 - Örneğin Microsoft Word'ün .doc ve .docx uzantılı dosyaları.
 - En önemli avantajı dosya üzerinde ileri ve geri gidilebilmesidir.

FILE

- C programlama dilinde giriş ve çıkışlar Akış(stream) adı verilen yapı üzerinden gerçekleştirilir.
- Bir dosya yazılıp okunmadan önce o dosya için bir akış oluşturulmalıdır.
- Bu akışı oluşturup takip edebilmek için FILE yapısı kullanılır.
- FILE genel kütüphanede tanımını bulunan bir yapıdır.
- Yani özetle bir dosya oluşturulmadan önce FILE tipinde bir işaretçi tanımlanmalıdır.

fopen

- Dosya üzerinde işlem yapmadan önce dosyanın açılması gereklidir.
- fopen fonksiyonu dosyayı açmaya yarar.
- Bu fonksiyon kullanılırken iki argüman alır: Açılacak dosya adı ve erişim modu. Erişim modu, hangi modda açılacağını tanımlar.
- `fopen("Yeni Metin Belgesi.txt", "w");`

Sembol	Erişim Modu
r	okuma (read)
w	yeniden yazma (write)
a	sona ekleme (append)
r+	okuma ve yazma (dosya zaten varolmalı)
w+	okuma ve yeniden yazma (değiştirme)
a+	okuma ve ekleme
rb	ikili dosyayı okuma amaçlı açar
r+b	ikili dosyayı okuma ve yazma amaçlı açar

fopen

- Eğer açma işlemi başarılıysa fopen fonksiyonu **dosya işaretçisi** döndürür. Başarısızsa null döndürür.
- Dosya işaretçisi tanımlamak ve yeni dosya oluşturmak için;
FILE *dosya_gostergesi;
dosya_gostergesi=fopen("Yeni Metin Belgesi.txt", "w");
- FILE tipi ve fopen gibi fonksiyonlar stdio.h kütüphanesinde tanımlıdır.
- fopen fonksiyonu işletim sistemine dosyanın kullanıldığını bildirir ve işletim sistemi o dosyanın başka programlar tarafından işlenmesine, silinmesine izin vermez

fopen

- Örnek-1:

The screenshot shows a C program in a code editor. The program includes `<stdio.h>` and defines a `main()` function. Inside `main()`, a file pointer `dosya_gostergesi` is declared and then assigned the result of `fopen("Deneme Metin Belgesi.txt", "w")`. The `return 0;` statement is at the end of the function. Below the code editor, a console window displays the output: "Process exited after 0.03829 seconds with return code 0" and "Press any key to continue . . .".

```
1  #include<stdio.h>
2  main() {
3      FILE * dosya_gostergesi;
4      dosya_gostergesi=fopen("Deneme Metin Belgesi.txt", "w");
5
6      return 0; }
7
```

Process exited after 0.03829 seconds with return code 0
Press any key to continue . . .

fclose

- Bir dosyayla işimiz bittiğinde o dosyayı kapatmamız gerekir.
- Bu bellek akışını fclose fonksiyonu ile sonlandırabiliriz.
- Kullanım:

`fclose (FILE *kapatilacak_dosya_adi);`

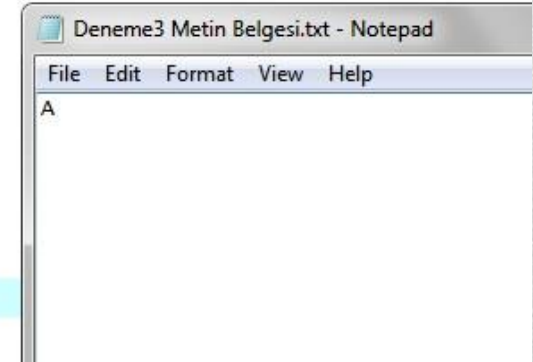
- Eğer dosya başarılı bir şekilde kapatıldıysa 0 değeri döndürür.
- Başarısız olursa sıfırdan farklı bir değer döndürülür.

```
FILE *dosya_gostergesi=fopen("Yeni Metin Belgesi.txt", "w");  
fclose(dosya_gostergesi)
```


Metin Dosyası İşlemleri

- Okuma ve yazma işlemlerini yapan fonksiyonlar 3 çeşittir:
 - karakter bazında işlem yapanlar: fputc, fgetc
 - biçimlendirilmiş metin bazında: fprintf, fscanf
 - satır bazında : fputs, fgets
- dosya_gostergesi ile işaret edilen bir dosyaya karakter yazdırma işlemi yapmak için;
fputc('z', dosya_gostergesi)
- Örnek-2:

```
1  #include<stdio.h>
2
3  main() {
4      FILE *dosya_gostergesi;
5      dosya_gostergesi=fopen("Deneme3 Metin Belgesi.txt", "w");
6      fputc('A', dosya_gostergesi);
7      return 0;
8  }
```



fputc, putc

- putc ve fputc birebir aynı işi yapmaktadırlar.
- fputc bir C fonksiyonuyken, putc ise bir makro olarak tanımlanmıştır.
- Performans gerektiren işlemlerde putc daha hızlı çalışır. Ancak makrolarla ilgili hatalar oluşabilmektedir. fputc kullanımı önerilir.
- Örnek-3:

```
1  #include<stdio.h>
2
3  main() {
4      int i;
5      char dizi1[]="Merhaba Dünya";
6      FILE *dosya_gostergesi;
7      dosya_gostergesi=fopen("Deneme4 Metin Belgesi.txt", "w");
8
9      for (i = 0; dizi1[i]!='\0'; i++)
10         fputc(dizi1[i], dosya_gostergesi);
11
12     return 0; }
13
```

C:\Users\Kemal\Desktop\Google Drive\masaustu\ders

```
-----
Process exited after 0.01004 seconds with return code 0
Press any key to continue . . .
```

Deneme4 Metin Belgesi - Notepad

File Edit Format View Help

Merhaba Dünya

EOF değeri ve feof fonksiyonu

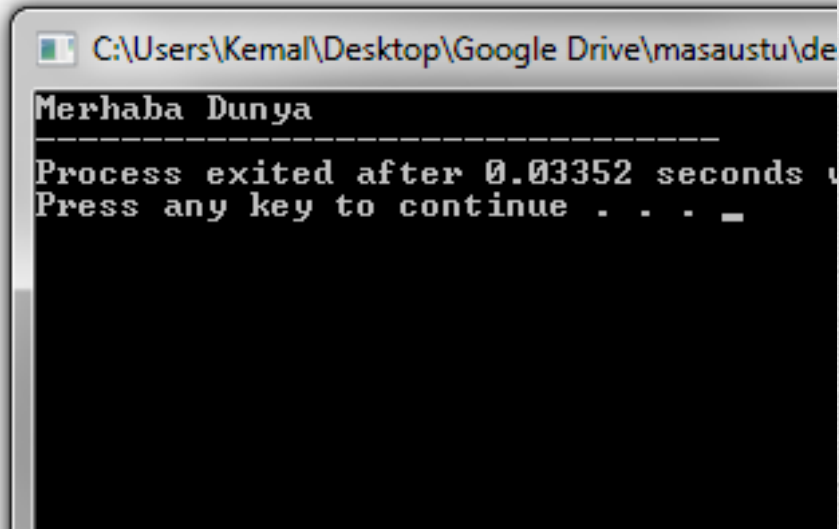
- EOF: end of file
- Bir dosyadan karakterlerin okunması işi yapılırken, dosyanın sonuna gelip gelmediğimizi gösteren bir göstergeye ihtiyaç duyarız.
- getc ve fgetc fonksiyonları dosyadan bir değer okunamadığında bir EOF değeri döndürürler.
- Dosya sonuna ulaşıldığında ya da dosyaya erişim bir sorundan dolayı kesildiğinde de EOF değeri döndürüldüğünden dolayı buna göre daha kullanışlı olan feof fonksiyonu üretilmiştir.
- Sadece dosyada okunacak bir karakter kalmadığında feof fonksiyonu sıfır dışında bir değer döndürür.

fgetc,getc

- Her iki fonksiyon da kendisine parametre olarak gelen dosyadan tek bir karakter okumaya yarar.
- Dosya yolu yazılmazsa programın bulunduğu klasör kastedilmiş olur.
- dosya=fopen("D:\\Deneme\\Deneme4 Metin Belgesi.txt", "r");
- Örnek-4a: EOF

while içinde okunan s değeri, EOF değilse döngü çalışmaya devam eder

```
1  #include<stdio.h>
2
3  main() {
4
5      FILE *dosya;
6      dosya=fopen("Deneme4 Metin Belgesi.txt", "r");
7      char s;
8      while((s=fgetc(dosya))!=EOF)
9      {
10         printf("%c",s);
11     }
12
13     fclose(dosya);
14     return 0; }
15
```



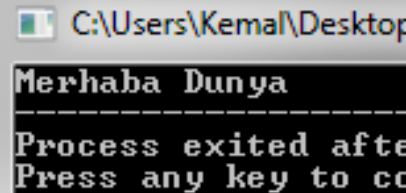
```
C:\Users\Kema\Desktop\Google Drive\masaustu\de
Merhaba Dünya
-----
Process exited after 0.03352 seconds
Press any key to continue . . . _
```


fgetc, getc

- Örnek-4b:

feof fonksiyonu sıfır dışında bir değer döndürdüğünde dosya sonuna gelinmiştir.

```
1  #include<stdio.h>
2  main() {
3      FILE *dosya;
4      dosya=fopen("Deneme4 Metin Belgesi.txt", "r");
5
6      if (dosya == NULL)    //Dosya açılmazsa çıkış yapılacak
7          return 0;
8
9      do
10     {
11         char c = fgetc(dosya); // Her seferinde tek karakter alacak
12
13         if (feof(dosya)) //Okunacak değer varsa sıfır değeri döndürülür
14             break ;
15
16         printf("%c", c);
17     } while(1);
18
19     fclose(dosya);
20     return 0; }
```



C:\Users\Kemal\Desktop
Merhaba Dünya

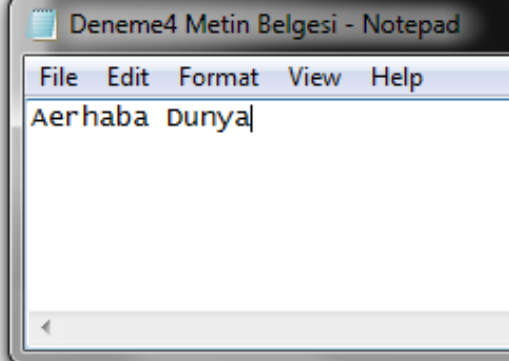
Process exited after
Press any key to co

fprintf ve fscanf

- Biçimlendirilmiş metin bazında işlem yapan bu iki fonksiyon aslında printf ve scanf gibi çalışır.
- fscanf'in farkı klavyeden okumak yerine dosyadan okumaktır.
- fprintf'in farkı ekrana yazmak yerine dosyaya yazmaktır.
- Belirtilen dosya akışı üzerinden çalışmaktadır.
- Örnek-5:

```
1  #include<stdio.h>
2  main() {
3      FILE *dosya1, *dosya2;
4
5      char ch;
6      dosya1=fopen("Deneme3 Metin Belgesi.txt", "r");
7      dosya2=fopen("Deneme4 Metin Belgesi.txt", "r+");
8      if (dosya1 == NULL) //Dosya açilamazsa çıkış yapılacak
9      {printf("dosya1 bulunamadi\n");
10      return 0;
11      }
12
13      fscanf(dosya1,"%c",&ch); // 1.dosyadaki A karakteri okunuyor
14
15      if (dosya2 == NULL) //Dosya açilamazsa çıkış yapılacak
16      {printf("dosya2 bulunamadi\n");
17      return 0;
18      }
19      fprintf(dosya2,"%c",ch); // 2.dosyanın başına A karakteri yazılıyor
20
21      fclose(dosya1);
22      fclose(dosya2);
23      return 0; }
24
```

Process exited after 0.01227
Press any key to continue .



fputs ve fgets

- Satır bazında işlem yapan bu iki fonksiyon putc ve getc gibi çalışmaktadır fakat sadece bir satır boyunca çalışırlar.
- Maksimum satır uzunluğu bilgisi satırdaki karakter sayısından fazla olsa bile alt satıra geçilmez.
- fgets(kelimeDizisi[i], maxSatirUzunlugu, dosya);
- fputs(kelimeDizisi, dosya1);
- Örnek-6:

```
1 #include<stdio.h>
2 main() {
3     FILE *dosya1, *dosya2;
4     char dizil[20];
5     int i;
6     dosya1=fopen("Deneme6 Metin Belgesi.txt", "w");
7
8     for(i=0;i<10;i++)
9     {
10        fprintf(dosya1,"%d",i); // dosyanın 1. satırına 0'dan 9'a kadar sayılar yazdırılıyor
11    }
12    fprintf(dosya1,"\n"); //Alt satıra geçiliyor
13    fputs("abcdefghij", dosya1); // dosyanın 2. satırına a'dan j'ye kadar harfler yazdırılıyor
14    fclose(dosya1);
15    dosya1=fopen("Deneme6 Metin Belgesi.txt", "r");
16    fgets(dizil, 20, dosya1);
17
18    printf("\nDosyanın 1. satiri:%s\n", dizil);
19
20    fclose(dosya1);
21
22    return 0; }
```

C:\Users\Kemal\Desktop\Google Drive\ma

Dosyanın 1. satiri:0123456789

Process exited after 0.008818 s
Press any key to continue . . .

Deneme6 Metin Belgesi - Notepad

File Edit Format View Help

0123456789
abcdefghij

rewind fonksiyonu

- Görevi imleci dosyanın başına getirmektir.
- Bir dosyada işlem yapılırken akış başlatıldığından işlemleri bu akış denetler. Bu akışa göre de dosya üzerinde okuma veya yazma işlemi yapıldığında, imleç görevin bittiği yerde kalır.
- Başa döndürmek istenilen dosyayı bu fonksiyona göndermek yeterlidir:
- Örnek:
`rewind(dosya1);`

fseek fonksiyonu

- Görevi imleci dosyanın belirli bir yerine götürmeye yarar.
- `int fseek(FILE *dosya, long offset, int lokasyon);`
- lokasyon için 3 seçenek mevcuttur:
 - `SEEK_SET` ise yeni konumu dosyanın başlangıcından itibaren ayarlar,
 - `SEEK_CUR` ise yeni konumu geçerli dosya konumundan itibaren ayarlar,
 - `SEEK_END` ise yeni konumu dosyanın sonundan itibaren ayarlar.
- Örnek:

`fseek (dosya, 10, SEEK_SET) //dosyanın en başından 10 bayt ileri`

`fseek (dosya, -5, SEEK_CUR) //imlecin aktif konumdan 5 bayt geri`

`fseek (dosya, -100, SEEK_END) //dosyanın en sonundan 100 bayt geri`

- İkili(binary) modda çalıştığı zaman işlevseldir. Metin modunda açıldığında ofset değeri için sadece 0 değeri kullanılabilir.

ftell fonksiyonu

- ftell fonksiyonu imlecin o an dosyanın neresinde olduğunu söyleyen bir fonksiyondur.
- Akış ikili (binary) modda açıldığı zaman bir anlamı vardır.
- `long int uzunluk = ftell (dosya);`

ÖDEV:

- Kullanıcıdan 5x5 lik bir matris girmesini isteyin.
- Bunu matris1.txt isimli bir dosyaya yazdırın.
- Ardından matris1'in transpozunu alıp matris2.txt isimli dosyaya yazın.
- Transpoz işlemini yaparken matris1.txt dosyasından her bir satırı veya karakteri okuyup ondan sonra matris2'ye yazdırın.

KAYNAKLAR

- Her Yönüyle C, Tefik KIZILÖREN, Kodlab