

Dosya Sistemi



**Suhap
SAHIN
Onur GÖK**

fopen()

```
FILE *fopen(const char *filename, const char *mode)
```

*filename: dosya yolu

*mode: açma modu

Text Dosyası

```
FILE *fopen(*dosya_yolu, *acma_modu)
```

```
// dosya_yolu
```

Windows

“d:\test.txt”

Text Dosyası

```
FILE *fopen(*dosya_yolu, *acma_modu)
```

// dosya_yolu

Windows

“d:\test.txt”

Linux

“/home/kullanici/test.txt”

Text Dosyası

FILE *fopen(*dosya_yolu, *acma_modu)

// acma_modu

- ❖ “r” : okuma
- ❖ “w” : yazma (dosyanın içeriğini silip baştan yazar, yoksa oluşturur)
- ❖ “a” : ekleme (dosya sonuna yazar, yoksa oluşturur)
- ❖ “r+” : okuma ve güncelleme (dosya yoksa açmaz, hata verir)
- ❖ “w+” : yazma ve güncelleme (dosyanın içeriğini silip açar)
- ❖ “a+” : ekleme ve güncelleme

ilk karakteri okuma

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

```
#include <stdlib.h>
```

```
int main() {
```

```
FILE * dosya = fopen("test.txt", "w");
```

```
// dosyayi ac
```

```
if (dosya == NULL) {
```

```
// dosyaya erisilemiyorsa NULL olur
```

```
printf("dosya acilamadi\n");
```

```
exit(1);
```

}

```
char c = fgetc(dosya);
```

```
if (c == EOF) {
```

```
// dosya sonu kontrolu
```

```
printf("dosya sonu, karakter yok\n");
```

// dosyanin sonuna gelindiyse EOF (-1) degeri okunur.

```
} else {
```

```
printf("okunan karakter: %c\n", c);
```

```
// dosyada ilk karakteri okunur.
```

}

```
fclose(dosya);
```

// dosyayi kapat

```
return 0;
```

}

birden fazla karakter okuma

```
#include <stdio.h>
#include <stdlib.h>
#define DOSYA_YOLU "test.txt"

int main() {

    FILE * dosya;
    char c;
    // dosyayi ac
    if ((dosya = fopen(DOSYA_YOLU, "r")) == NULL) {
        printf("dosya acilamadi!\n");
        exit(1);
    }

    // dosya sonuna gelene kadar dongu calissin
    while ((c = fgetc(dosya)) != EOF) {
        printf("karakter: %c (%d)\n", c, c);
    }

    // dosyayi kapat
    fclose(dosya);
    return 0;
}
```

fgets()

`char *fgets(char *str, int n, FILE *stream)`

`*str`: Okunan “string”i tutan işaretçi

`n`: Okunacak maksimum karakter sayısı(boş karakter dahil)

`*stream`: Okunacak string

fgets()

`char *fgets(char *str, int n, FILE *stream)`

İşlem başarılı ise, *str işaretçisi döner

Dosya Sonu ile karşılaşıldığında ve/veya hiçbir karakter okunmazsa, *str işaretçisinin içeriği değişmeden kalır ve boş işaretçisi (Null Pointer) döndürülür.

Bir hata oluşursa, boş işaretçi(Null Pointer) döner.

formatlı okuma

```
#include <stdio.h>
#include <stdlib.h>
#define DOSYA_YOLU "test.txt"

int main() {
    satir_satir_oku();
    printf("\n-----\n");
    formatli_oku();
    return 0;
}
```

formatlı okuma

```
void satir_satir_oku() {  
    FILE * dosya;  
    char buf[100];  
  
    if ((dosya = fopen(DOSYA_YOLU, "r")) == NULL) {  
        printf("dosya acilamadi!\n");  
        exit(1);  
    }  
  
    int satir_sayisi = 0;  
    // satir satir okuma islemi  
    while ( fgets(buf, 100, dosya) != NULL ) {  
        satir_sayisi++;  
        printf("%d. satir: \"%s\"\n", satir_sayisi, buf);  
    }  
}
```

fscanf()

```
int fscanf(FILE *stream, const char *format, char *buf)
```

*stream: String'in okunduğu yeri tutan işaretçi

*buf: Okunan string'in saklandığı yeri tutan işaretçi

*format: Okunan karakterin nasıl tutulacağı

formatlı okuma

```
void formatli_oku() {  
    FILE * dosya;  
  
    if ((dosya = fopen(DOSYA_YOLU, "r")) == NULL) {  
        printf("dosya acilamadi!\n");  
        exit(1);  
    }  
  
    char buf[100];  
  
    fscanf(dosya, "%99s", buf); // max 99 harfli kelime oku  
    printf("okunan kelime: %s\n", buf);  
  
    fscanf(dosya, "%99s", buf);  
    printf("okunan kelime: %s\n", buf);  
  
    fclose(dosya);  
}
```

Dosyada okunacak yeri belirleme

```
#include <stdio.h>
#include <stdlib.h>
#define DOSYA_YOLU "test.txt"
int main() {
    FILE * dosya;
    if ((dosya = fopen(DOSYA_YOLU, "r")) == NULL) {
        printf("dosya acilamadi!\n");
        exit(1);
    }
    fseek(dosya, 0, SEEK_END);
    long int len = ftell(dosya);
    printf("dosyanin boyutu: %ld byte\n", len);
    fseek(dosya, 0, SEEK_SET);
    long int gosterge = ftell(dosya);
    printf("%ld. indexteki karakter okunacak\n", gosterge);
    char c = fgetc(dosya);
    printf("okunan karakter: %c\n\n", c);
    gosterge = ftell(dosya);
    printf("%ld. indexteki karakter okunacak\n", gosterge);
    c = fgetc(dosya);
    printf("okunan karakter: %c\n\n", c);
    fseek(dosya, len-2, SEEK_SET);
    c = fgetc(dosya);
    printf("sondan iki onceki karakter: %c\n", c);
    fclose(dosya);
    return 0;
}
```

// dosyanin sonuna atla
// gostergecin bulundugu byte numarasini oku

// dosya basina atla
// bulundugu byte numarasini oku

// bulundugu byte numarasini oku

// sondan iki onceki karaktere atla

// dosyayi kapat

text dosyasına yazmak

```
#include <stdio.h>
#include <stdlib.h>
#define DOSYA_YOLU "test2.txt"
int main() {
    FILE * dosya;
    if ((dosya = fopen(DOSYA_YOLU, "w")) == NULL) {
        printf("dosya acilamadi!\n");
        exit(1);
    }
    fputc('a', dosya);
    fputc('1', dosya);
    fputc('2', dosya);
    fclose(dosya);
    if ((dosya = fopen(DOSYA_YOLU, "a")) == NULL) {
        printf("dosya acilamadi!\n");
        exit(1);
    }
    long int gosterge = ftell(dosya);
    printf("%ld. indexteki karakter yazilacak\n", gosterge);
    fputs("satir 1\n", dosya);
    fputs("...\n", dosya);
    fprintf(dosya, "%06d %.3f\n", 15, 1.3);
    fprintf(dosya, "test\n");
    fclose(dosya);
    return 0;
}
```

// dosyayi bastan yazmak icin ac (dosyanin icindekileri siler)

// a12

karakterlerini tek tek yaz

// dosyayi kapat

// dosyayi eklemek icin ac

// bulunduğu byte numarasını oku

// satir satir yaz

// formatli yaz

// dosyayi kapat

Binary dosya(r, w)-Struct-

```
#include <stdio.h>
#include <stdlib.h>
#define DOSYA_YOLU "kayit.bin"
```

```
int main() {
```

```
    int secim;
```

```
    // Bundan sonra yazılacaklar bu araya yazılacak
```

```
    return 0;
```

```
}
```


Binary dosya(r, w)-Struct-

```
while (1) {  
    printf("\n1: listele\n");  
    printf("2: kayit ekle\n");  
    printf("3: kayitlari sil\n");  
    printf("0: cikis\n");  
    printf("secim?: ");  
    scanf("%d", &secim);  
  
    switch (secim) {  
        case 0:  
            exit(0);  
        case 1:  
            listele();  
            break;  
        case 2: {  
            struct Ogrenci ogr;  
            printf("no ad soyad not?:");  
            scanf("%s %s %s %d", ogr.no, ogr.ad, ogr.soyad, &ogr.not);  
            kayit_ekle(ogr);  
            break;  
        }  
        case 3: {  
            kayitlari_sil();  
            break;  
        }  
        default:  
            printf("yalis secim\n");  
    };  
};
```

Binary dosya(r, w)-Struct-

```
struct Ogrenci {  
    char no[10];  
    char ad[20];  
    char soyad[20];  
    int not;  
};
```

fread(), fwrite()

`size_t fread(void *ptr, size_t size, size_t nmemb, FILE *stream)`

`size_t fwrite(void *ptr, size_t size, size_t nmemb, FILE *stream)`

*ptr: yazılacak/okunacak olan eleman

size: byte cinsinden boyut(sizeof)

nmemb: Eleman sayısı

*stream: yazılacak/okunacak olan dosya

Binary dosya(r, w)-Struct-

```
void kayit_ekle(struct Ogrenci ogr) {  
    FILE * dosya;  
  
    // dosyayi ekleme modunda ac  
    if ((dosya = fopen(DOSYA_YOLU, "a")) == NULL) {  
        printf("dosya acilamadi!\n");  
        exit(1);  
    }  
  
    fwrite(&ogr, sizeof(struct Ogrenci), 1, dosya);  
  
    fclose(dosya);  
}
```

Binary dosya(r, w)-Struct-

```
void listele() {  
    FILE * dosya;  
  
    // dosyayi okuma modunda ac  
    if ((dosya = fopen(DOSYA_YOLU, "r")) == NULL) {  
        printf("dosya acilamadi!\n");  
        exit(1);  
    }  
    printf("\nliste:\n");  
    while (1) {  
        struct Ogrenci ogrenci;  
  
        size_t sayi = fread(&ogrenci, sizeof(struct Ogrenci), 1, dosya);  
        if (sayi < 1)  
            break; // okunamadiysa (dosya bitmisse vb.) donguyu durdur  
        printf("%s %s %s %d\n", ogrenci.no, ogrenci.ad, ogrenci.soyad, ogrenci.not);  
    }  
  
    fclose(dosya);  
}
```

Binary dosya(r, w)-Struct-

```
void kayitlari_sil() {  
    FILE * dosya;  
  
    // dosyayi yazma modunda ac  
    if ((dosya = fopen(DOSYA_YOLU, "w")) == NULL) {  
        printf("dosya acilamadi!\n");  
        exit(1);  
    }  
  
    // kapat  
    fclose(dosya);  
}
```

Sorular

