

# ALGORİTMA VE PROGRAMLAMA II

## HAFTA#7

# Genel Bakış...

2

- Dosya İşlemleri
  - ASCII Tablosu ve Karakterler
  - fgetc() ve fputc() Fonksiyonları
  - feof() Fonksiyonu

# 6. BÖLÜM

3

## Dosya İşlemleri - Bölüm 2

# ASCII Tablosu

4

- Bilgisayarın belirttiğimiz
  - sayıları,
  - harfleri,
  - noktalama işaretlerini
  - vb. tüm karakterleri işleyebilmesi için belirli code'lara ihtiyacı bulunmaktadır.
- **ASCII** (*American Standard Code for Information Interchange*) tablosu bu kodları bize sağlar.

ASCII value	Character	Control character	ASCII value	Character	ASCII value	Character	ASCII value	Character
0	(null)	NUL	32	(space)	64	@	96	
1	☺	SOH	33	!	65	A	97	a
2	☹	STX	34	"	66	B	98	b
3	♥	ETX	35	#	67	C	99	c
4	♦	EOT	36	\$	68	D	100	d
5	♣	ENQ	37	%	69	E	101	e
6	♠	ACK	38	&	70	F	102	f
7	(beep)	BEL	39	'	71	G	103	g
8	■	BS	40	(	72	H	104	h
9	(tab)	HT	41	)	73	I	105	i
10	(line feed)	LF	42	*	74	J	106	j
11	(home)	VT	43	+	75	K	107	k
12	(form feed)	FF	44	,	76	L	108	l
13	(carriage return)	CR	45	-	77	M	109	m
14	♪	SO	46	.	78	N	110	n
15	☼	SI	47	/	79	O	111	o
16	▲	DLE	48	0	80	P	112	p
17	▼	DC1	49	1	81	Q	113	q
18	↕	DC2	50	2	82	R	114	r
19	!!	DC3	51	3	83	S	115	s
20	π	DC4	52	4	84	T	116	t
21	§	NAK	53	5	85	U	117	u
22	▬	SYN	54	6	86	V	118	v
23	↕	ETB	55	7	87	W	119	w
24	↑	CAN	56	8	88	X	120	x
25	↓	EM	57	9	89	Y	121	y
26	→	SUB	58	:	90	Z	122	z
27	←	ESC	59	;	91	[	123	{
28	(cursor right)	FS	60	<	92	\	124	
29	(cursor left)	GS	61	=	93	]	125	}
30	(cursor up)	RS	62	>	94	^	126	~
31	(cursor down)	US	63	?	95	_	127	☐

# ASCII Tablosu (devam...)

6

## Örnek:

- **char** tipinde bir değişken tanımlayarak ‘a’ değerini atayınız.
- Ekrana ‘a’ nın karşılık geldiği ASCII sayı kodunu yazdırınız.
- Ekrana bu sayıyı kullanarak nasıl ‘f’ yazdırınız.
- Ekrana gülen yüz 😊 karakterini yazdırınız.

# ASCII Tablosu (devam...)

7

```
int main()
{
    char x;

    x = 'a';
    printf("%d \n", x);
    printf("%c \n", x);

    x = x + 5;
    printf("%d \n", x);
    printf("%c\n", x);

    printf("Gulen yuz: %c\n", 2);

    return 0;
}
```

# fgetc() ve fputc() Fonksiyonları

8

- Genel yapıları aşağıdaki gibidir:  
`int fgetc (FILE *fp);`  
`int fputc (int id, FILE *fp);`
- `fputc()` fonksiyonu `id` isimli değişken değeri olan byte'ı `fp` ile gösterilen dosyaya `char` olarak yazar. **id değişkeni int bir değer olarak tanımlandığı halde**, `char` bir değer olarak **çağırabilirsiniz (ASCII değeri)**.
- Dosyaya yazma işlemi başarılıysa, `fputc()` fonksiyonu yazılan karakteri değilse **EOF** değerini geri verir.



# fgetc() ve fputc() Fonksiyonları (devam...)

9

- `fgetc()` fonksiyonu, `fp` ile işaret edilen dosyadaki bir sonraki byte'ı `char` olarak okur ve `int` bir değer olarak geri verir.
- `fgetc()` fonksiyonu bir hata durumunda ve dosya sonuna geldiğinde `int` bir değer olan `EOF` değerini geri verdiği için, geri verilen değer kontrolü amacıyla `fgetc()` fonksiyonu `int` bir değer geri verir.
- Ancak, `fgetc()` fonksiyonunun geri verdiği değeri `int` bir değişkene atamanız şart değildir. Geri verilen değeri bir karakter değişkene de atayabilirsiniz.

# Örnek 1: fputc ve fgetc Kullanımı

10

- fputc() fonksiyonunu kullanarak, 'a' karakterini dosyaya yazınız.
- fgetc() fonksiyonunu kullanarak 'a' karakterini dosyadan okuyunuz ve ekrana yazdırınız.

# Örnek 1: fputc ve fgetc Kullanımı

11

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

if (fp == NULL)
{
    printf("Dosya olusturulurken hata olustu...\n");
    exit(1);
}
fputc('a', fp);
fclose(fp);

fp = fopen("dosya.txt", "r");
if (fp == NULL)
{
    printf("Dosya olusturulurken hata olustu...\n");
    exit(1);
}
```

# Örnek 1: fputc ve fgetc Kullanımı

12

```
char okunan;  
okunan = fgetc(fp);  
if (okunan == EOF)  
    printf("Karakter okunamadi...\n");  
else  
    printf("Okunan karakter:%d", okunan);  
  
fclose(fp);  
  
return 0;  
}
```

# Örnek 2: `fputc` ve `fgetc` Kullanımı

13

- `fputc()` fonksiyonunu kullanarak, her defasında bir karakter olmak üzere, 10 adet 'a' harfini (w) modunda açılan dosyaya yazar.
- Bu işlemi bitirdikten sonra, dosyayı kapatır.
- Dosyayı (r) modunda açtıktan sonra, her karakteri birer birer dosyadan okuyarak ekrana yazar.

# Örnek 2: fputc ve fgetc Kullanımı

14

```
#include "stdio.h"

int main()
{
    FILE *fp;
    int i;

    if ((fp = fopen ("dosya.txt", "w")) == NULL)
    {
        printf("Dosya açma hatası!");
        exit(1);
    }

    for (i=0; i<10; i++)
        fputc ('a', fp);

    fclose(fp);

    if ((fp = fopen ("dosya.txt", "r")) == NULL)
    {
        printf("Dosya açma hatası!");
        exit(1);
    }

    for (i=0; i<10; i++)
        printf("%c", fgetc(fp));
    fclose(fp);

    return 0;
}
```

# Örnek 2: `fputc` ve `fgetc` Kullanımı

15

- For döngüsü kullanmadan dosyadan okuma yapabilir miyiz?
- Deneyelim...

# Örnek 3: fputc ve fgetc Kullanımı

16

- Klavyeden girilen bir karakter dizisini önce **dizi** adlı bir karakter dizisine atalım.
- Sonra, dizi içeriğini (w) modu ile açtığı dosya1.txt adlı bir metin dosyasına yazıp ve dosyayı kapatalım.
- Dosyayı bu kez sadece okumak için (r) modunda açıp, dosya içeriğini okuyup ekrana yazdıktan sonra tekrar dosyayı kapatalım.



# Örnek 3: fputc ve fgetc Kullanımı - 1

17

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

void main (void)
{
    char dizi[41];
    FILE *fp;
    char *cp;
    int i;

    printf("Bir karakter dizisi giriniz (En fazla 40 karakter): ");
    gets(dizi);

    if ((fp = fopen("dosya1.txt", "w")) == NULL) {
        printf("Dosya açma hatası!\n");
        exit(1);
    }

    cp = dizi;
    while (*cp) {
        if (fputc (*cp, fp) == EOF) {
            printf("Dosyaya yazma hatası!\n");
            exit(1);
        }
        cp++;
    }
    fclose (fp);
}
```

# Örnek 3: fputc ve fgetc Kullanımı - 2

18

```
if ((fp = fopen("dosya1.txt", "r")) == NULL) {  
    printf("Dosya açma hatası!\n");  
    exit(1);  
}  
  
for ( ; ; ) {  
    i = fgetc(fp);  
    if (i == EOF) break;  
    putchar(i);  
}  
  
fclose (fp);
```

# feof() Fonksiyonu

19

- feof() fonksiyonu yapısı aşağıdaki gibidir:

```
int feof (FILE *fp);
```

- Eğer `fp` ile gösterilen dosyanın sonuna gelinmişse, `feof()` fonksiyonu 0 olmayan bir değer, aksi takdirde 0 değerini geri verir.

# feof() Fonksiyonu (devam...)

20

- `fgetc()` fonksiyonu, aşağıda belirtilen 2 farklı durumda, EOF değerini geri verir.
  1. Bir hata meydana geldiğinde
  2. Dosya sonuna gelindiğinde

**DİKKAT:** feof fonksiyonunu kullanırken sonsuz döngü durumuna düşebilirsiniz. Döngü içerisinde mutlaka ekstra kontrol yapmalısınız.

# Örnek 4: `feof()` Kullanımı

21

- `feof()` fonksiyonu kullanarak bir dosyanın byte cinsinden boyutunu hesaplayalım.

# Örnek 4: feof() Kullanımı

22

```
void main ()
{
    FILE *fp;
    int udl;

    if ((fp = fopen("dosya.txt", "rb")) == NULL) {
        printf("Dosya acilamadi!\n");
        exit(1);
    }
    udl = 0;
    while (!feof(fp))
    {
        if (fgetc(fp) != EOF)
            udl++;
        if (ferror(fp))
        {
            printf("Dosyadan okuma hatasi!\n");
            exit(1);
        }
    }
    printf("Verilen dosya uzunlugu: %d byte\n", udl);
    fclose (fp);
}
```

# Örnek 4: `feof()` Kullanımı

23

- `feof()` fonksiyonu **kullanmadan** bir dosyanın byte cinsinden boyutunu hesaplayalım.

# Örnek 4: feof() Kullanımı

24

```
while (1)
{
    if (fgetc(fp) != EOF)
        udl++;
    else
        break;
    if (ferror(fp))
    {
        printf("Dosyadan okuma hatasi!\n");
        exit(1);
    }
}
printf("Verilen dosya uzunlugu: %d byte\n", udl);
fclose (fp);
```



# Bazı Dosya Uygulamaları

25

1. Dosyada harf arama ve kaç tane olduğunu bulma.
2. Dosyadaki bir harfi başka bir harf ile değiştirme.
3. Bir dosyanın içeriğini başka bir dosyaya kopyalama.
4. Dosyada her harfin kaç defa geçtiğini bulup, çoktan aza doğru sıralama.
5. Dosyanın şifrelenmesi, şifresinin çözülmesi.

# KAYNAKLAR

26

- N. Ercil Çağıltay ve ark., C DERSİ PROGRAMLAMAYA GİRİŞ, Ada Matbaacılık, ANKARA; 2009.
- Milli Eğitim Bakanlığı "Programlamaya Giriş ve Algoritmalar Ders Notları", 2007
- C Programlama Dili, Şerafettin ARIKAN
- Problem Solving and Program Design in C, Hanly, Koffman
- <http://www.AlgoritmaveProgramlama.com>



Algoritma ve Programlama

# İYİ ÇALIŞMALAR...