

C Programlama Dili

UNIX altında derlenmiş örnek programların çıktıları

Hazırlayan Ismet Kocaman

1 / 29

```
Sistem      : Solaris 2.5/ SunSparc Ultra-1
               uname -a çıktısı:
               SunOS username 5.5.1 Generic_103640-08 sun4u sparc SUNW,Ultra-1
Derleme    : cc -o program_adi program_adi.c
Çalıştırma: # program_adi <enter>
```

BOLUM 2

*** * * Örnek 2.8-1 tip.c programı**

integral promotion

OK !

c << 3 --> Sonuç 0 değildir

c << 3 --> Sonuç 0

k1 : 100, k2 : -56, k3 : 44

k1 : 0x64, k2 : 0xFFFFFC8, k3 : 0x2C

uk1 : 100, uk2 : 200, uk3 : 44

uk1 : 0x64, uk2 : 0xC8, uk3 : 0x2C

k1 : 0x7C

k1 + k1 : 0xF8

k2 : 0xFFFFFFFF8

BOLUM 4

*** * * Örnek 4.2-5 dosyaa.c programı**

DOSYAA

i : 10, j : 20

Not:

Programı oluşturmak için dosyab.c ile birleştirilecek.

(Derleme: cc -o dosyaa dosyaa.c dosyab.c)

*** * * Örnek 4.5-1 bagla1.c programı**

BAGLA1

12.000000

123

13

C Programlama Dili

UNIX altında derlenmiş örnek programların çıktıları

Hazırlayan ***İsmet Kocaman***

2 / 29

45

Not:

Programı oluşturmak için bagla2.c programı ile birleştirilecek.

(Derleme: cc [diğer opsiyonlar...] -o bagla1 bagla1.c bagla2.c)

BOLUM 5

*** * * Örnek 5.1-1 adresop1.c**

Dizi[0] için bellek adresi : 4026530788

Dizi[1] için bellek adresi : 4026530792

Dizi[2] için bellek adresi : 4026530796

*** * * Örnek 5.1-2 adresop2.c**

n değiskeni : 4 byte, bellek adresi : 4026530796

adres değeri : 4 byte.

*** * * Örnek 5.1-3 adresop3.c**

adresler: &n : 4026530796, &k : 4026530792

değerler: p : 4026530796, *p : 1650, n : 1650, k : 1650

p : 4, n : 4, k : 4 byte

*** * * Örnek 5.1-4 adresop4.c**

int : 4, long : 4, double : 8

int * : 4, long * : 4, double * : 4

&i : 4026530796, &j : 4026530792

i : 10, j : 40, pi : 4026530796, *pi : 10

i : 89, j : 40, pi : 4026530796, *pi : 89

i : 42, j : 40, pi : 4026530796, *pi : 42

i : 42, j : 40, pi : 4026530792, *pi : 40

*** * * Örnek 5.2-1 dizi1.c**

Dizi[0] : 0, bellek adresi : 4026530788

Dizi[1] : 1, bellek adresi : 4026530792

Dizi[2] : 2, bellek adresi : 4026530796

*** * * Örnek 5.2-2 dizi2.c**

Dizi[0] : 0, bellek adresi : 4026530788

Dizi[1] : 1, bellek adresi : 4026530792

Dizi[2] : 2, bellek adresi : 4026530796

C Programlama Dili

UNIX altında derlenmiş örnek programların çıktıları

Hazırlayan Ismet Kocaman

3 / 29

*** * * Örnek 5.2-3 arit1.c**

```
sizeof(char) : 1
sizeof(int) : 4
sizeof(double) : 8
(char *)0 + 1 : 1
(int *)0 + 1 : 4
(double *)0 + 1 : 8
```

*** * * Örnek 5.2-4 arit2.c**

```
short i : 2 byte, pi : 4026530798, pi + 1 : 4026530800
long l : 4 byte, pl : 4026530792, pl + 1 : 4026530796
double x : 8 byte, px : 4026530784, px + 1 : 4026530792
&i : 4026530798, &i + 1 : 4026530800
&l : 4026530792, &l + 1 : 4026530796
&x : 4026530784, &x + 1 : 4026530792
```

*** * * Örnek 5.2-5 dizi3.c**

```
*(pDizi + 0) : 0, bellek adresi pDizi + 0 : 4026530788
*(pDizi + 1) : 1, bellek adresi pDizi + 1 : 4026530792
*(pDizi + 2) : 2, bellek adresi pDizi + 2 : 4026530796
```

*** * * Örnek 5.2-6 dizi4.c**

```
dizi elemanlarının adresleri
&Dizi[0] : 4026530788, &Dizi[0] + 0 : 4026530788, Dizi + 0 : 4026530788
&Dizi[1] : 4026530792, &Dizi[0] + 1 : 4026530792, Dizi + 1 : 4026530792
&Dizi[2] : 4026530796, &Dizi[0] + 2 : 4026530796, Dizi + 2 : 4026530796
dizi elemanlarının değerleri
Dizi[0] : 0, *(Dizi + 0) : 0
Dizi[1] : 1, *(Dizi + 1) : 1
Dizi[2] : 2, *(Dizi + 2) : 2
dizi elemanlarının adresleri
pDizi + 0 : 4026530788
pDizi + 1 : 4026530792
pDizi + 2 : 4026530796
dizi elemanlarının değerleri
pDizi[0] : 0, *(pDizi + 0) : 0
pDizi[1] : 1, *(pDizi + 1) : 1
pDizi[2] : 2, *(pDizi + 2) : 2
pDizi = Dizi + 1 deyimi sonrası değerler
```

C Programlama Dili

UNIX altında derlenmiş örnek programların çıktıları

Hazırlayan Ismet Kocaman

4/29

Dizi : 4026530788, pDizi : 4026530792, *pDizi : 1

pDizi - Dizi : 1

pDizi++ deyimi sonrası değerler

Dizi : 4026530788, Dizi + 2 : 4026530796, pDizi : 4026530796, *pDizi : 2

pDizi > Dizi

pDizi == Dizi + 2

*** * * Örnek 5.2-8 adr.c**

p : 4026530788, *p : 99, i : 0

i = *p++

p : 4026530792, *p : 88, i : 99

p : 4026530792, *p : 88, i : 99

i = *++p

p : 4026530796, *p : 77, i : 77

p : 4026530796, *p : 77, i : 77

i = ++*p

p : 4026530796, *p : 78, i : 78

p : 4026530796, *p : 78, i : 78

i = (*p)++

p : 4026530796, *p : 79, i : 78

0, 1, 2, 3, 4,

7, 8, 3, 9, 5,

*** * * Örnek 5.3-1 karak.c**

sizeof a : 5, sizeof b : 5

strlen(a): 4, strlen(b): 4

210b8

210c0

023

*** * * Örnek 5.4-1 adrdizgi.c**

sizeof AdrDizgi : 16, sizeof *AdrDizgi : 4

sizeof **AdrDizgi : 1

sizeof AdrDizgi[0] : 4, sizeof AdrDizgi[0][0] : 1

AdrDizgi : effffbe0, *AdrDizgi : 213bc

**AdrDizgi : 97

AdrDizgi[0] : 213bc, AdrDizgi[0][0] : 97

AdrDizgi [0] : 213bc, *AdrDizgi [0] : a, AdrDizgi[0] dizgisi : a

AdrDizgi [1] : 213be, *AdrDizgi [1] : b, AdrDizgi[1] dizgisi : bcd987

C Programlama Dili

UNIX altında derlenmiş örnek programların çıktıları

Hazırlayan Ismet Kocaman

5 / 29

AdrDizgi [2] : 213c5, *AdrDizgi [2] : 1, AdrDizgi[2] dizgisi : 123

a

bcd987

123

a

bcd987

123

AdrDizgi[0] dizgisi 1 karakter.

AdrDizgi[1] dizgisi 6 karakter.

AdrDizgi[2] dizgisi 3 karakter.

	Adres	Karakter	ASCII kod
	-----	-----	-----
AdrDizgi[0][0] :	213bc	a	97
AdrDizgi[0][1] :	213bd		0
AdrDizgi[1][0] :	213be	b	98
AdrDizgi[1][1] :	213bf	c	99
AdrDizgi[1][2] :	213c0	d	100
AdrDizgi[1][3] :	213c1	9	57
AdrDizgi[1][4] :	213c2	8	56
AdrDizgi[1][5] :	213c3	7	55
AdrDizgi[1][6] :	213c4		0
AdrDizgi[2][0] :	213c5	1	49
AdrDizgi[2][1] :	213c6	2	50
AdrDizgi[2][2] :	213c7	3	51
AdrDizgi[2][3] :	213c8		0

*** * * Örnek 5.5-1 ciftadr.c programı**

d : 9.3, *pd : 9.3, **ppd : 9.3

&d : effffbe8, pd : effffbe8, *ppd : effffbe8

&pd : effffbe4, ppd : effffbe4

*** * * Örnek 5.6-1 adrcevir.c programı**

cDizi dizisi 8 byte.

1.byte : 0xA1

2.byte : 0xB2

3.byte : 0xC3

4.byte : 0xD4

5.byte : 0xA0

C Programlama Dili

UNIX altında derlenmiş örnek programların çıktıları

Hazırlayan Ismet Kocaman

6 / 29

6.byte : 0xB0

7.byte : 0xC0

8.byte : 0xD0

*Adrc : 0xA0

Tams : 0xA0B0C0D0

NOT:

adrcevir.c programında aşağıdaki değişiklikler yapıldı:

...

```
unsigned char cDizi[] = { 0xA1, 0xB2, 0xC3, 0xD4,  
                          0xA0, 0xB0, 0xC0, 0xD0 },
```

...

```
Adrc = (unsigned char *) ( (unsigned int *)Adrc + 1 );
```

...

```
Tams = *(unsigned *) (Adrc + 4);
```

```
printf( "Tams : 0x%08X\n", Tams );
```

...

*** * * Örnek 5.6-2 void.c programı**

c : A

i : 0xABCD, &i : effffbe8

effffbe8 adresinde bulunan byte : 0x00

effffbe9 adresinde bulunan byte : 0x00

effffbea adresinde bulunan byte : 0xAB

effffbeb adresinde bulunan byte : 0xCD

*** * * Örnek 5.7-5 size.c programı**

p adres degiskeni : 4 byte

s karakter dizisi : 4 byte

123 dizgisi : 4 byte

*** * * Örnek 5.7-11 adrf11.c**

pd[0] : effffbc8, *pd[0] : 3.0, d[0] : 3.0

pd[1] : effffbd0, *pd[1] : 1.9, d[1] : 1.9

pd[2] : effffbd8, *pd[2] : 2.8, d[2] : 2.8

pd[3] : effffbe0, *pd[3] : 5.7, d[3] : 5.7

pd[4] : effffbe8, *pd[4] : 4.6, d[4] : 4.6

pd[0] : effffbd0, *pd[0] : 1.9, d[0] : 3.0

pd[1] : effffbd8, *pd[1] : 2.8, d[1] : 1.9

pd[2] : effffbc8, *pd[2] : 3.0, d[2] : 2.8

C Programlama Dili

UNIX altında derlenmiş örnek programların çıktıları

Hazırlayan Ismet Kocaman

7 / 29

```
pd[3] : effffbe8, *pd[3] : 4.6, d[3] : 5.7
pd[4] : effffbe0, *pd[4] : 5.7, d[4] : 4.6
```

*** * * Örnek 5.7-12 adrf12.c programı**

```
pd[0] : effffbc8, *pd[0] : 3.0, d[0] : 3.0
pd[1] : effffbd0, *pd[1] : 1.9, d[1] : 1.9
pd[2] : effffbd8, *pd[2] : 2.8, d[2] : 2.8
pd[3] : effffbe0, *pd[3] : 5.7, d[3] : 5.7
pd[4] : effffbe8, *pd[4] : 4.6, d[4] : 4.6
pd[0] : effffbd0, *pd[0] : 1.9, d[0] : 3.0
pd[1] : effffbd8, *pd[1] : 2.8, d[1] : 1.9
pd[2] : effffbc8, *pd[2] : 3.0, d[2] : 2.8
pd[3] : effffbe8, *pd[3] : 4.6, d[3] : 5.7
pd[4] : effffbe0, *pd[4] : 5.7, d[4] : 4.6
```

*** * * Örnek 5.7-13 adrf13.c programı**

```
pd[0] : effffbc8, *pd[0] : 3.0, d[0] : 3.0
pd[1] : effffbd0, *pd[1] : 1.9, d[1] : 1.9
pd[2] : effffbd8, *pd[2] : 2.8, d[2] : 2.8
pd[3] : effffbe0, *pd[3] : 5.7, d[3] : 5.7
pd[4] : effffbe8, *pd[4] : 4.6, d[4] : 4.6
pd[0] : effffbd0, *pd[0] : 1.9, d[0] : 3.0
pd[1] : effffbd8, *pd[1] : 2.8, d[1] : 1.9
pd[2] : effffbc8, *pd[2] : 3.0, d[2] : 2.8
pd[3] : effffbe8, *pd[3] : 4.6, d[3] : 5.7
pd[4] : effffbe0, *pd[4] : 5.7, d[4] : 4.6
```

*** * * Örnek 5.7-20 arg1.c programı**

```
argc : 3
```

```
argv[0] :          arg1,  4 karakter
```

```
argv[1] :          arg1,  4 karakter
```

```
argv[2] :          arg2,  4 karakter
```

```
NOT:
```

```
arg1.c programında argv[i] için null check yaparak NULL degerin
printf'e gitmesini onlemek gerekiyor. Aksi taktirde program
calisirken "Segmentation Fault" hatasi olusur. (DOS'ta olusmaz).
```

```
...
```

```
printf("argc : %d\n", argc );
```

C Programlama Dili

UNIX altında derlenmiş örnek programların çıktıları

Hazırlayan **İsmet Kocaman**

8 / 29

```
for ( i = 0; i <= argc; i++ )
if ( argv[i] != '\0' )
    printf("argv[%d] : %12s, %2u karakter\n",
i, argv[ i ], strlen( argv[i] ) );
...
```

*** * * Örnek 5.7-21 arg2.c programı**

arg2 123 abc <enter> komutu ile çıktı:

```
while - 1
123
abc
while - 2
argv[...] --> 4026531117
4026531117 adresindeki karakter --> 1
4026531118 adresindeki karakter --> 2
4026531119 adresindeki karakter --> 3
argv[...] --> 4026531121
4026531121 adresindeki karakter --> a
4026531122 adresindeki karakter --> b
4026531123 adresindeki karakter --> c
while - 3
argv[...] --> 4026531117
4026531117 adresindeki karakter --> 1
4026531118 adresindeki karakter --> 2
4026531119 adresindeki karakter --> 3
argv[...] --> 4026531121
4026531121 adresindeki karakter --> a
4026531122 adresindeki karakter --> b
4026531123 adresindeki karakter --> c
for - 1
arg2 dizgisi 4 karakter
123 dizgisi 3 karakter
abc dizgisi 3 karakter
for - 2
arg2 dizgisi 4 karakter
123 dizgisi 3 karakter
abc dizgisi 3 karakter
```


C Programlama Dili

UNIX altında derlenmiş örnek programların çıktıları

Hazırlayan Ismet Kocaman

9 / 29

```
while - 5
123
abc
while - 6
1
a
while - 7
1
a
while - 4
123
abc
sizeof argv : 4, sizeof *argv : 4, sizeof **argv : 1
```

NOT:

arg2.c programında "Segmentation Fault" hatası almamak için aşağıdaki null-check deyimleri eklendi...

(Bu hata puts fonksiyonuna NULL aktarılınca oluşur. Fakat DOS'ta aynı hata oluşmaz)

```
...
puts( "for - 1" );
...
argv = av;
for ( i = 0; i <= argc; i++ )
    if ( *(argv + i) != 0 )
        printf( "%s dizgisi %u karakter\n",
                *(argv + i), strlen( argv[i] ) );

...
puts( "for - 2" );
...
for ( i = 0; i <= argc; i++, argv++ )
{
    if ( *argv != 0 )
        printf( "%s dizgisi %u karakter\n",
                *argv, strlen( *argv ) );
    ...
}
```

C Programlama Dili

UNIX altında derlenmiş örnek programların çıktıları

Hazırlayan Ismet Kocaman

10 / 29

*** * * Örnek 5.7-22 arg3.c programı**

```
arg3
123
abc
arg3
123
abc
12ab
```

*** * * bits.c programı**

```
Tarih1 : C79F, (00000000000000001100011110011111)
Tarih2 : FE630000, (11111110011000110000000000000000)
Tarih2 - Gun 31, Ay 12, Yil 1999
Tarih2 - Gun 31, (11111)
Tarih2 - Ay 12, (1100)
Tarih2 - Yil 99, (1100011)
```

*** * * endian.c programı**

```
i : 0xABCD, &i : effffbe8, (00000000000000001010101111001101)
```

```
Adres: Deger: Hex, (Binary)
```

```
-----
effffbe8      0x00, (00000000)
effffbe9      0x00, (00000000)
effffbea      0xAB, (10101011)
effffbeb      0xCD, (11001101)
n : 0xAB12CD34, &n : effffbe0, (10101011000100101100110100110100)
Adres: Deger: Hex, (Binary)
```

```
-----
effffbe0      0xAB, (10101011)
effffbe1      0x12, (00010010)
effffbe2      0xCD, (11001101)
effffbe3      0x34, (00110100)
```

NOT:

endian.c programında yukarıdaki çıktıyı almak için aşağıdaki
degisiklikler yapıldi:

Satir 12 ve 25'de (char *) yerine (unsigned char *) yerlestirildi.

*** * * sizeof.c programı**

```
n : 4 byte, j : 12 byte, jj : 12 byte
```

C Programlama Dili

UNIX altında derlenmiş örnek programların çıktıları

Hazırlayan Ismet Kocaman

11 / 29

```
-sizeof- s : 4 byte, t : 5 byte
-strlen- s : 4 karakter, t : 4 karakter
char : 1, int : 4, long : 4 byte
j dizisinin eleman sayısı : 3
jj dizisinin eleman sayısı : 3
j dizisi int[3] tipindedir : 12 byte
-sizeof- t dizisinin eleman sayısı : 5
sizeof 2L : 4 byte, sizeof 2 : 4 byte
sizeof "1234" : 5 byte
strlen("1234") : 4 karakter
* * * varunix.c programı
abc
def
xyz
```

BOLUM 6

*** * * Örnek 6.1-1 adr2b.c programı**

```
satir, sutun: deger(adres)
0,0: 0 (ffffffbb0) 0,1: 1 (ffffffbb4) 0,2: 2 (ffffffbb8) 0,3: 3 (ffffffbbc) 0,4: 4 (ffffffbc0)
1,0: 5 (ffffffbc4) 1,1: 6 (ffffffbc8) 1,2: 7 (ffffffbcc) 1,3: 8 (ffffffbd0) 1,4: 9 (ffffffbd4)
2,0: 2 (ffffffbd8) 2,1: 3 (ffffffbdc) 2,2: 4 (ffffffbe0) 2,3: 5 (ffffffbe4) 2,4: 6 (ffffffbe8)
3,0: 7 (ffffffbec) 3,1: 8 (ffffffbf0) 3,2: 9 (ffffffbf4) 3,3: 0 (ffffffbf8) 3,4: 1 (ffffffbfc)
iDizi2b          : fffffffb0, sizeof(iDizi2b)          : 80
iDizi2b + 1       : fffffffc4, sizeof(iDizi2b + 1)     : 4
iDizi2b[ 1 ]      : fffffffc4, sizeof(iDizi2b[ 1 ])    : 20
iDizi2b[ 1 ] + 1  : fffffffc8, sizeof(iDizi2b[ 1 ] + 1) : 4
iDizi2b[ 1 ][ 1 ] : 6,      sizeof(iDizi2b[ 1 ][ 1 ]) : 4
Dizi elemanlarının degerlerinin listelenmesi :
*( iDizi2b[ satir ] + sutun ) :
0 1 2 3 4
5 6 7 8 9
2 3 4 5 6
7 8 9 0 1
sizeof(Adr2b) : 4, sizeof(*Adr2b) : 20
Adr2b : fffffffb0, Adr2b + 1 : fffffffc4
```

C Programlama Dili

UNIX altında derlenmiş örnek programların çıktıları

Hazırlayan Ismet Kocaman

12/29

Adr2b[satir][sutun] :

```
0 1 2 3 4
5 6 7 8 9
2 3 4 5 6
7 8 9 0 1
```

*** * * Örnek 6.1-2 adr3b.c programı**

Tablo, Satir, Sutun: Deger(Adres) :

Tablo No : 0

```
0,0,0: 0(ffffffb10) 0,0,1: 1(ffffffb14) 0,0,2: 2(ffffffb18) 0,0,3: 3(ffffffb1c) 0,0,4: 4(ffffffb20)
0,1,0: 5(ffffffb24) 0,1,1: 6(ffffffb28) 0,1,2: 7(ffffffb2c) 0,1,3: 8(ffffffb30) 0,1,4: 9(ffffffb34)
0,2,0: 2(ffffffb38) 0,2,1: 3(ffffffb3c) 0,2,2: 4(ffffffb40) 0,2,3: 5(ffffffb44) 0,2,4: 6(ffffffb48)
0,3,0: 7(ffffffb4c) 0,3,1: 8(ffffffb50) 0,3,2: 9(ffffffb54) 0,3,3: 0(ffffffb58) 0,3,4: 1(ffffffb5c)
```

Tablo No : 1

```
1,0,0: 0(ffffffb60) 1,0,1: 1(ffffffb64) 1,0,2: 2(ffffffb68) 1,0,3: 3(ffffffb6c) 1,0,4: 4(ffffffb70)
1,1,0: 5(ffffffb74) 1,1,1: 6(ffffffb78) 1,1,2: 7(ffffffb7c) 1,1,3: 8(ffffffb80) 1,1,4: 9(ffffffb84)
1,2,0: 2(ffffffb88) 1,2,1: 3(ffffffb8c) 1,2,2: 4(ffffffb90) 1,2,3: 5(ffffffb94) 1,2,4: 6(ffffffb98)
1,3,0: 7(ffffffb9c) 1,3,1: 8(ffffffba0) 1,3,2: 9(ffffffba4) 1,3,3: 0(ffffffba8) 1,3,4: 1(ffffffbac)
```

Tablo No : 2

```
2,0,0: 0(ffffffbb0) 2,0,1: 1(ffffffbb4) 2,0,2: 2(ffffffbb8) 2,0,3: 3(ffffffbbc) 2,0,4: 4(ffffffbc0)
2,1,0: 5(ffffffbc4) 2,1,1: 6(ffffffbc8) 2,1,2: 7(ffffffbcc) 2,1,3: 8(ffffffbd0) 2,1,4: 9(ffffffbd4)
2,2,0: 2(ffffffbd8) 2,2,1: 3(ffffffbdc) 2,2,2: 4(ffffffbe0) 2,2,3: 5(ffffffbe4) 2,2,4: 6(ffffffbe8)
2,3,0: 7(ffffffbec) 2,3,1: 8(ffffffbf0) 2,3,2: 9(ffffffbf4) 2,3,3: 0(ffffffbf8) 2,3,4: 1(ffffffbfc)
```

ifade : degeri, sizeof (ifade)

iDizi3b : fffffffb10, 240

iDizi3b + 1 : fffffffb60, 4

iDizi3b[1] : fffffffb60, 80

iDizi3b[1] + 1 : fffffffb74, 4

iDizi3b[1][1] : fffffffb74, 20

iDizi3b[1][1] + 1 : fffffffb78, 4

iDizi3b[1][1][1] : 6, 4

Dizi elemanlarının degerlerinin listelenmesi:

*(*(*(iDizi3b + i) + j) + k) :

Tablo No : 0

```
0 1 2 3 4
5 6 7 8 9
```

C Programlama Dili

UNIX altında derlenmiş örnek programların çıktıları

Hazırlayan Ismet Kocaman

13 / 29

2 3 4 5 6

7 8 9 0 1

Tablo No : 1

0 1 2 3 4

5 6 7 8 9

2 3 4 5 6

7 8 9 0 1

Tablo No : 2

0 1 2 3 4

5 6 7 8 9

2 3 4 5 6

7 8 9 0 1

sizeof(Adr3b) : 4, sizeof(*Adr3b) : 80

Adr3b : effffffb10, Adr3b + 1 : effffffb60

Adr3b[tablo][satir][sutun]:

0 1 2 3 4

5 6 7 8 9

2 3 4 5 6

7 8 9 0 1

0 1 2 3 4

5 6 7 8 9

2 3 4 5 6

7 8 9 0 1

0 1 2 3 4

5 6 7 8 9

2 3 4 5 6

7 8 9 0 1

*** * * Örnek 6.1-3 adr4b.c programı**

0 1 2 3 4

5 6 7 8 9

2 3 4 5 6

7 8 9 0 1

0 1 2 3 4

5 6 7 8 9

2 3 4 5 6

7 8 9 0 1

C Programlama Dili

UNIX altında derlenmiş örnek programların çıktıları

Hazırlayan Ismet Kocaman

14 / 29

```
0 1 2 3 4
5 6 7 8 9
2 3 4 5 6
7 8 9 0 1
```

```
0 1 2 3 4
5 6 7 8 9
2 3 4 5 6
7 8 9 0 1
0 1 2 3 4
5 6 7 8 9
2 3 4 5 6
7 8 9 0 1
0 1 2 3 4
5 6 7 8 9
2 3 4 5 6
7 8 9 0 1
```

*** * * Örnek 6.2-1 fonkdizi.c programı**

sizeof iDizi2b : 80

- 2b dizi listeleme -

sizeof Adr2b : 4, sizeof *Adr2b : 20, sizeof *Adr2b / sizeof (int) : 5

```
0 1 2 3 4
5 6 7 8 9
2 3 4 5 6
7 8 9 0 1
```

sizeof iDizi3b : 240

- 3b dizi listeleme -

sizeof Adr3b : 4, sizeof *Adr3b : 80, sizeof *Adr3b / sizeof (int) : 20

```
0 1 2 3 4
5 6 7 8 9
2 3 4 5 6
7 8 9 0 1
0 1 2 3 4
5 6 7 8 9
```

C Programlama Dili

UNIX altında derlenmiş örnek programların çıktıları

Hazırlayan Ismet Kocaman

15 / 29

```
2 3 4 5 6
7 8 9 0 1
0 1 2 3 4
5 6 7 8 9
2 3 4 5 6
7 8 9 0 1
```

sizeof iDizi4b : 480

- 4b dizi listeleme -

sizeof Adr4b : 4, sizeof *Adr4b : 240, sizeof *Adr4b / sizeof (int) : 60

```
0 1 2 3 4
5 6 7 8 9
2 3 4 5 6
7 8 9 0 1
0 1 2 3 4
5 6 7 8 9
2 3 4 5 6
7 8 9 0 1
0 1 2 3 4
5 6 7 8 9
2 3 4 5 6
7 8 9 0 1
```

```
0 1 2 3 4
5 6 7 8 9
2 3 4 5 6
7 8 9 0 1
0 1 2 3 4
5 6 7 8 9
2 3 4 5 6
7 8 9 0 1
0 1 2 3 4
5 6 7 8 9
2 3 4 5 6
7 8 9 0 1
```

C Programlama Dili

UNIX altında derlenmiş örnek programların çıktıları

Hazırlayan Ismet Kocaman

16/29

2b dizinin 1. satırı ile 2. satirinin değiştirilmesi:

- 2b dizi listeleme -

sizeof Adr2b : 4, sizeof *Adr2b : 20, sizeof *Adr2b / sizeof (int) : 5

5 6 7 8 9

0 1 2 3 4

2 3 4 5 6

7 8 9 0 1

BOLUM 7

*** * * Örnek 7.1-1 din2b3b.c programı**

2-b dizi :

8 8 3 5 1

7 0 9 2 6

9 7 4 0 5

3 9 3 7 6

3-b dizi :

6 8 5 1 7

4 1 5 2 6

4 5 4 2 1

7 4 3 1 6

8 8 5 3 7

6 5 3 2 4

3 0 4 3 2

8 2 7 7 5

9 3 1 4 0

9 8 7 9 6

6 0 8 6 4

2 6 5 1 4

*** * * Örnek 7.1-2 din2b.c programı**

8 8 3

5 1 7

8 8 3

5 1 7

C Programlama Dili

UNIX altında derlenmiş örnek programların çıktıları

Hazırlayan Ismet Kocaman

17 / 29

*** * * Örnek 7.1-3 din3b.c programı**

```
8 8 3 5
1 7 0 9
2 6 9 7
4 0 5 3
9 3 7 6
6 8 5 1
8 8 3 5
1 7 0 9
2 6 9 7
4 0 5 3
9 3 7 6
6 8 5 1
```

*** * * Örnek 7.1-4 dinfonk.c programı**

Satır Sayısı ? 5

Sütun Sayısı ? 20

```
8 8 3 5 1 7 0 9 2 6
9 7 4 0 5 3 9 3 7 6
6 8 5 1 7 4 1 5 2 6
4 5 4 2 1 7 4 3 1 6
8 8 5 3 7 6 5 3 2 4
```

*** * * Örnek 7.1-5 remalloc.c programı**

214b8 : 20 byte bellek alanı ayrıldı !!!

24c88 : 40 byte bellek alanı ayrıldı !!!

24cb8 : 60 byte bellek alanı ayrıldı !!!

24d00 : 80 byte bellek alanı ayrıldı !!!

0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18,

NOT:

Programda aşağıdaki cast gerekli.

...

Adr = (int *)realloc(Adr, ToplamByte(i), ToplamByte(Onceki));

...

*** * * Örnek 7.1-6 realloc.c programı**

21380 : 20 byte bellek alanı ayrıldı !!!

24b50 : 40 byte bellek alanı ayrıldı !!!

24b50 : 60 byte bellek alanı ayrıldı !!!

C Programlama Dili

UNIX altında derlenmiş örnek programların çıktıları

Hazırlayan Ismet Kocaman

18/29

24b50 : 80 byte bellek alanı ayrıldı !!!

0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18,

NOT:

Programda aşağıdaki cast gerekli.

...

Adr = (int *)realloc(Adr, ToplamByte(i));

...

*** * * Örnek 7.1-7 adrdindz.c programı**

Random Dizgi --> mzhrlajoetbkwltztvie

AdrDinDz [0] : mzhrlajoetbkwltztvie

Random Dizgi --> hpfybpswajccvjof

AdrDinDz [1] : hpfybpswajccvjof

Random Dizgi --> eeebjzylneexk

AdrDinDz [2] : eeebjzylneexk

Random Dizgi --> zgasxfvxffucryjdugcmkk

AdrDinDz [3] : zgasxfvxffucryjdugcmkk

Random Dizgi --> ixjetobwkvdqj

AdrDinDz [4] : ixjetobwkvdqj

Random Dizgi --> hxatzivobcto

AdrDinDz [5] : hxatzivobcto

AdrDinDz [2] : eeebjzylneexk

AdrDinDz [2] [3] : b

*AdrDinDz : mzhrlajoetbkwltztvie

(*AdrDinDz)[4] : l

*(AdrDinDz + 4) : l

*** * * Örnek 7.2-1 dinstru.c programı**

Kayıt Yapısı :

(*p_s).dizgi : XyZabc123K, (*p_s).i : 99

p_s[0].dizgi : XyZabc123K, p_s[0].i : 99

Kayıt Yapi Dizisi :

XyZabc123K 1

XyZabc123K 2

XyZabc123K 3

XyZabc123K 4

XyZabc123K 5

2-b Yapi Dizisi :

C Programlama Dili

UNIX altında derlenmiş örnek programların çıktıları

Hazırlayan Ismet Kocaman

19 / 29

XyZabc123K 2 XyZabc123K 2 XyZabc123K 2

XyZabc123K 2 XyZabc123K 3 XyZabc123K 4

2-b adres dizisi ile 2-b Yapi Dizisine erişim :

XyZabc123K 2 XyZabc123K 2 XyZabc123K 2

XyZabc123K 2 XyZabc123K 3 XyZabc123K 4

NOT:

Örnek 7.2-1 dinstru.c programında satır 210'da Kayit** yerine

Kayit*** bulunacak.

```
...
for ( i = 0; i < NoSatir2b; i++ )
    if ( (p_ap_2as[ i ] = (Kayit **)malloc( NoSutun2b * sizeof (Kayit*) ))
        == NULL )

return (Kayit***)NULL;
...
```

BOLUM 8

*** * * Örnek 8.1-1 yapı1.c programı**

p1.isim: xyz, p1.basla: 1991, p1.son: 1997

p1 : 12 byte

p1.isim : 4 byte

p1.basla : 4 byte

p1.son : 4 byte

p2.isim: xyz, p2.basla: 1991, p2.son: 1997

p2 : 20 byte

p2.isim : 11 byte

p2.basla : 4 byte

p2.son : 4 byte

*** * * Örnek 8.2-4 yapidiz4.c programı**

t.s1 : 5, t.s2 : 5, p.s1 : 4, p.s2 : 4 byte

t.s1 : abc, t.s2 : xyz, p.s1 : abc, p.s2 : xyz

t.s1[0] : a, t.s1[1] : b, t.s1[2] : c, t.s1[3] : , t.s1[4] : ,

t.s2[0] : x, t.s2[1] : y, t.s2[2] : z, t.s2[3] : , t.s2[4] : ,

p.s1[0] : a, p.s1+0 : 135300

p.s1[1] : b, p.s1+1 : 135301

C Programlama Dili

UNIX altında derlenmiş örnek programların çıktıları

Hazırlayan Ismet Kocaman

20 / 29

```
p.s1[2] : c, p.s1+2 : 135302
p.s2[0] : x, p.s2+0 : 135304
p.s2[1] : y, p.s2+1 : 135305
p.s2[2] : z, p.s2+2 : 135306
```

*** * * Örnek 8.3-1 adryapi1.c programı**

```
sizeof(struct proje *) : 4, sizeof(ap) : 4, sizeof(p): 12 byte
ap->isim : yeni proje, ap->basla : 1991, ap->son : 1994
(*ap).isim : yeni proje, (*ap).basla : 1991, (*ap).son : 1994
(&p)->isim : yeni proje, (&p)->basla : 1991, (&p)->son : 1994
YENİ PROJE dizgisi ve 1990 değeri atandıktan sonra ...
```

```
p.isim : YENİ PROJE, p.basla : 1990, p.son : 1994
```

```
proje isminin 3. karakteri : N
```

```
bellek adresleri :
```

```
&p : 4026530788, &(p.isim) : 4026530788, &(p.basla) : 4026530792, &(p.son) : 4026530796
```

```
ap : 4026530788, &(ap->isim) : 4026530788, &(ap->basla) : 4026530792, &(ap->son) : 4026530796
```

NOT:

Programda tüm %d'ler %u formata çevrildi...

*** * * Örnek 8.3-3 adryapi3.c programı**

```
sizeof( p ) : 48, sizeof( adr_dizi ) : 12
```

```
yapi elemanları :
```

```
abc, xxx, 123 456, 23
```

```
def, yyy, 789 012, 34
```

```
mno, zzz, 890 123, 30
```

```
a
```

```
b
```

```
c
```

```
d
```

```
e
```

```
f
```

```
m
```

```
n
```

```
o
```

*** * * Örnek 8.3-4 adryapi4.c programı**

```
YDizi'nin eleman sayısı : 4
```

```
YDizi'nin bir elemanının byte sayısı : 12
```

C Programlama Dili

UNIX altında derlenmiş örnek programların çıktıları

Hazırlayan Ismet Kocaman

21 / 29

* * * Örnek 8.3-5 adryapi5.c programı

```
YDizi[0].FAdi : fonk1 --> fonksiyon 1 ...
YDizi[1].FAdi : fonk2 --> fonksiyon 2 ...
fonksiyon adresleri :
    fonk1 : 10b48, fonk2 : 10b64, fonk3 : 10b80
eleman adresleri :
    YDizi[0].adr_fonk : 10b48, YDizi[1].adr_fonk : 10b64
atama sonrası eleman adresleri :
    YDizi[0].adr_fonk : 10b48, YDizi[1].adr_fonk : 10b80
YDizi[1].FAdi : fonk3 --> fonksiyon 3 ...
```

* * * Örnek 8.3-6 adryapi6.c programı

```
elemanların değerleri ve bellek adresleri:
    i : 1, f : 0.7, c : A, adr : dizgi
    &s : 4026530776, &s.i : 4026530776, &s.f : 4026530784, &s.c : 4026530792, &s.adr : 4026530796
elemanların -hesaplanan- offset değerleri :
    s.i : 0, s.f : 4, s.c : 12, s.adr : 13 byte.
s yapısının -hesaplanan- byte sayısı : 17 byte.
adr elemanının -hesaplanan- bellek adresi : 4026530789
elemanların gerçek offset değerleri (offsetof makrosu):
    s.i : 0, s.f : 8, s.c : 16, s.adr : 20 byte.
s yapısı : 24 byte.
adr elemanının bellek adresi : 4026530796
NOT:
Programdaki tüm %d'ler %u formata çevrilmiştir.
```

* * * Örnek 8.5-1 union1.c programı

```
55
27.91
dizgi
bilinmeyen union tipi : 4
1077668085
```

* * * Örnek 8.5-2 union2.c programı

```
YDizi[0].UnDeg.j : 27
YDizi[1].UnDeg.j : 42
*YDizi[0].UnDeg.c : a
YDizi[1].UnDeg.c[0] : x
```

C Programlama Dili

UNIX altında derlenmiş örnek programların çıktıları

Hazırlayan Ismet Kocaman

22 / 29

*** * * Örnek 8.6-1 bitalan.c programı**

BitDeg.alan1 : 1

BitDeg.alan2 : 0

BitDeg.alan3 : 1

BOLUM 9

*** * * Örnek 9.1-1 textyaz.c programı**

NOT:

test.asc dosyasını oluşturur. Aşağıdaki değişiklik yapıldı:

...

if ((fp = fopen("test.asc", "w")) == NULL)

...

ls -l test.asc

-rw-rw-r-- 1 ismet staff 7 Mar 25 12:43 test.asc

*** * * Örnek 9.1-2 textoku.c programı**

NOT:

...

if ((fp = fopen("test.asc", "r")) == NULL)

...

*** * * Örnek 9.1-3 mod1.c programı**

test.asc: r - text modunda okuma.

d 0x64, 100

e 0x65, 101

n 0x6E, 110

e 0x65, 101

m 0x6D, 109

e 0x65, 101

0x0A, 10

EOF (end-of-file) : -1

test.asc: rb - binary modunda okuma.

d 0x64, 100

e 0x65, 101

n 0x6E, 110

e 0x65, 101

m 0x6D, 109

C Programlama Dili

UNIX altında derlenmiş örnek programların çıktıları

Hazırlayan Ismet Kocaman

23 / 29

```
e 0x65, 101
  0x0A, 10
EOF (end-of-file) : -1
# ls -l test.asc
-rw-rw-r--  1 ismet    staff          7 Mar 25 12:43 test.asc
```

#

DOS Çıktısı !!!

```
/*
 * Çıktı:
 *
 * test.asc: r - text modunda okuma.
 * d 0x64, 100
 * e 0x65, 101
 * n 0x6E, 110
 * e 0x65, 101
 * m 0x6D, 109
 * e 0x65, 101
 *   0x0A, 10
 * EOF (end-of-file) : -1
 * test.asc: rb - binary modunda okuma.
 * d 0x64, 100
 * e 0x65, 101
 * n 0x6E, 110
 * e 0x65, 101
 * m 0x6D, 109
 * e 0x65, 101
 *   0x0D, 13  <----- !!!
 *   0x0A, 10
 * EOF (end-of-file) : -1
 */
```

NOT:

UNIX'te text ve binary mod farkı yoktur.
"a:\\test.asc" --> "test.asc"

```
...
if ( !(fp = fopen( "a:\\test.asc", "r" )) )
```

C Programlama Dili

UNIX altında derlenmiş örnek programların çıktıları

Hazırlayan Ismet Kocaman

24 / 29

```
...
if ( !(fp = fopen( "a:\\test.asc", "rb" )) )
...

* * * Örnek 9.1-4 mod2.c programı

# cat text
abc
123
def
# cat binary1
abc
123
def
# cat binary2
abc
123
def
# dump
Dosya adı ? : text
Hex (Dec ) ----- Hexadecimal Kodlar ----- --- ASCII Kod ----
000B(0011) 61 62 63 0A 31 32 33 0A 64 65 66          |abc.123.def      |
Toplam byte: 11
# dump
Dosya adı ? : binary1
Hex (Dec ) ----- Hexadecimal Kodlar ----- --- ASCII Kod ----
000B(0011) 61 62 63 0A 31 32 33 0A 64 65 66          |abc.123.def      |
Toplam byte: 11
# dump
Dosya adı ? : binary2
Hex (Dec ) ----- Hexadecimal Kodlar ----- --- ASCII Kod ----
000D(0013) 61 62 63 0D 0A 31 32 33 0D 0A 64 65 66    |abc..123..def    |
Toplam byte: 13
# ls -l *
-rw-rw-r--  1 ismet  staff      11 Mar 25 12:35 binary1
-rw-rw-r--  1 ismet  staff      13 Mar 25 12:35 binary2
-rw-rw-r--  1 ismet  staff      11 Mar 25 12:35 text
#
```


C Programlama Dili

UNIX altında derlenmiş örnek programların çıktıları

Hazırlayan Ismet Kocaman

25 / 29

DOS Çıktısı !!!

A:\> type text

abc

123

def

A:\> type binary1

abc

123

def

A:\> type binary2

abc

123

def

A:\> dump

Dosya adı ? : text

Hex (Dec)	-----	Hexadecimal Kodlar	-----	--- ASCII Kod ---
000D(0013)	61 62 63 0D 0A 31 32 33 0D 0A 64 65 66			abc...123..def

Toplam byte: 13

A:\> dump

Dosya adı ? : binary1

Hex (Dec)	-----	Hexadecimal Kodlar	-----	--- ASCII Kod ---
000B(0011)	61 62 63 0A 31 32 33 0A 64 65 66			abc.123.def

Toplam byte: 11

A:\> dump

Dosya adı ? : binary2

Hex (Dec)	-----	Hexadecimal Kodlar	-----	--- ASCII Kod ---
000D(0013)	61 62 63 0D 0A 31 32 33 0D 0A 64 65 66			abc...123..def

Toplam byte: 13

*** * * Örnek 9.1-7 bin.c programı**

dizi[0] : 790, 0x000000316

dizi[1] : 791, 0x000000317

dizi[2] : 792, 0x000000318

dizi[3] : 793, 0x000000319

dizi[4] : 794, 0x00000031A

dizi[5] : 795, 0x00000031B

C Programlama Dili

UNIX altında derlenmiş örnek programların çıktıları

Hazırlayan Ismet Kocaman

26 / 29

```
dizi[6] : 796, 0x0000031C
dizi[7] : 797, 0x0000031D
dizi[8] : 798, 0x0000031E
dizi[9] : 799, 0x0000031F
```

```
00 00 03 16 00 00 03 17 00 00 03 18 00 00 03 19 00 00 03 1A 00 00 03 1B 00 00 03 1C 00 00 03 1D 00 00 03 1E 00 00 03 1F
```

```
ls -l sayi.bin çıktısı:
```

```
-rw-rw-r-- 1 ismet staff 40 Mar 25 08:22 sayi.bin
```

```
DOS çıktısı !!! :
```

```
16 03 17 03 18 03 19 03 1A 03 1B 03 1C 03 1D 03 1E 03 1F 03
```

NOT:

Format 0x04X yerine UNIX'te 0x08X olmalıdır.

```
...
```

```
printf("dizi[%d] : %4d, 0x%08X\n", i, a_dizi[i], a_dizi[i] );
```

```
...
```

*** * * Örnek 9.2-1 hatatest.c programı**

NOT:

```
...
```

```
#include <errno.h>
```

```
...
```

```
# ls -l
```

```
-rw-rw-r-- 1 ismet staff 8 Mar 25 13:20 deneme
-rw-rw-r-- 1 ismet staff 17 Mar 25 13:20 VERI.DAT
-rw-rw-r-- 1 ismet staff 96 Mar 25 13:19 veriler
```

```
#
```

*** * * Örnek 9.2-2 error.c programı**

```
# cat error.log
```

```
errno : 0
```

```
errno : 2
```

```
#
```

*** * * Örnek 9.3-1 dump.c programı**

```
# ls -l sayi.bin
```

```
-rw-rw-r-- 1 ismet staff 40 Mar 25 08:22 sayi.bin
```

```
# dump
```

```
Dosya adı ? : <enter>
```

C Programlama Dili

UNIX altında derlenmiş örnek programların çıktıları

Hazırlayan Ismet Kocaman

27 / 29

Dosya adı girilmedi.

Kullanım: dump <dosya listesi>

dump sayi.bin

Dosya adı : sayi.bin

Hex (Dec)	-----	Hexadecimal Kodlar	-----	--- ASCII Kod ---
0010(0016)	00 00 03 16 00 00 03 17 00 00 03 18 00 00 03 19			
0020(0032)	00 00 03 1A 00 00 03 1B 00 00 03 1C 00 00 03 1D			
0028(0040)	00 00 03 1E 00 00 03 1F			

Toplam byte: 40

dump

Dosya adı ? : sayi.bin

Hex (Dec)	-----	Hexadecimal Kodlar	-----	--- ASCII Kod ---
0010(0016)	00 00 03 16 00 00 03 17 00 00 03 18 00 00 03 19			
0020(0032)	00 00 03 1A 00 00 03 1B 00 00 03 1C 00 00 03 1D			
0028(0040)	00 00 03 1E 00 00 03 1F			

Toplam byte: 40

dump sayi.bin xxx

Dosya adı : sayi.bin

Hex (Dec)	-----	Hexadecimal Kodlar	-----	--- ASCII Kod ---
0010(0016)	00 00 03 16 00 00 03 17 00 00 03 18 00 00 03 19			
0020(0032)	00 00 03 1A 00 00 03 1B 00 00 03 1C 00 00 03 1D			
0028(0040)	00 00 03 1E 00 00 03 1F			

Toplam byte: 40

Dosya adı : xxx

xxx acilamadi.

dump *.bin

Dosya adı : sayi.bin

Hex (Dec)	-----	Hexadecimal Kodlar	-----	--- ASCII Kod ---
0010(0016)	00 00 03 16 00 00 03 17 00 00 03 18 00 00 03 19			
0020(0032)	00 00 03 1A 00 00 03 1B 00 00 03 1C 00 00 03 1D			
0028(0040)	00 00 03 1E 00 00 03 1F			

Toplam byte: 40

tty

/dev/pts/4

dump /dev/pts/4

Dosya adı : /dev/pts/4

Hex (Dec)	-----	Hexadecimal Kodlar	-----	--- ASCII Kod ---
------------	-------	--------------------	-------	-------------------

C Programlama Dili

UNIX altında derlenmiş örnek programların çıktıları

Hazırlayan Ismet Kocaman

28 / 29

deneme

^D

0007(0007) 64 65 6E 65 6D 65 0A

|deneme.

|

Toplam byte: 7

#

*** * * Örnek 9.3-2 say.c programı**

say

Kullanım : say -opsiyon [dosya_listesi]

Opsiyonlar: (k)elime, ka(r)akter, (s)atır

Örnek : say -krs say.c

say -krs

deneme

^D

ls -l say*

-rw-rw-r-- 1 ismet staff 20 Mar 25 13:04 sayilar.bin

-rw-rw-r-- 1 ismet staff 85 Mar 25 13:04 sayilar.txt

*** * * Örnek 9.4-2 harfcev.c programı**

NOT:

...

if ((fp = fopen("VERI.DAT", "wb")) == NULL)

...

if ((fp = fopen("VERI.DAT", "rb+")) == NULL)

...

*** * * crnl.c programı**

i : 2000

i : 4999

NL : 0x0A, CR : 0x0D

NL : 10, CR : 13

Deneyiniz !!!

CRNL > CRNdosya

cat CRNdosya

*** * * filtre.c programı**

filtre <enter>

C Programlama Dili

UNIX altında derlenmiş örnek programların çıktıları

*Hazırlayan **Ismet Kocaman***

29 / 29

Bu Bir dENEMedir <enter>

bu bir denemedir

^D

Copyright © Ismet Kocaman. Her Hakkı Saklıdır.