

BLM111 Programlama Dilleri I

Hafta 11

Karakter Tutan Diziler

Mehmet Zahid YILDIRIM

Katar (String) Tanımlama

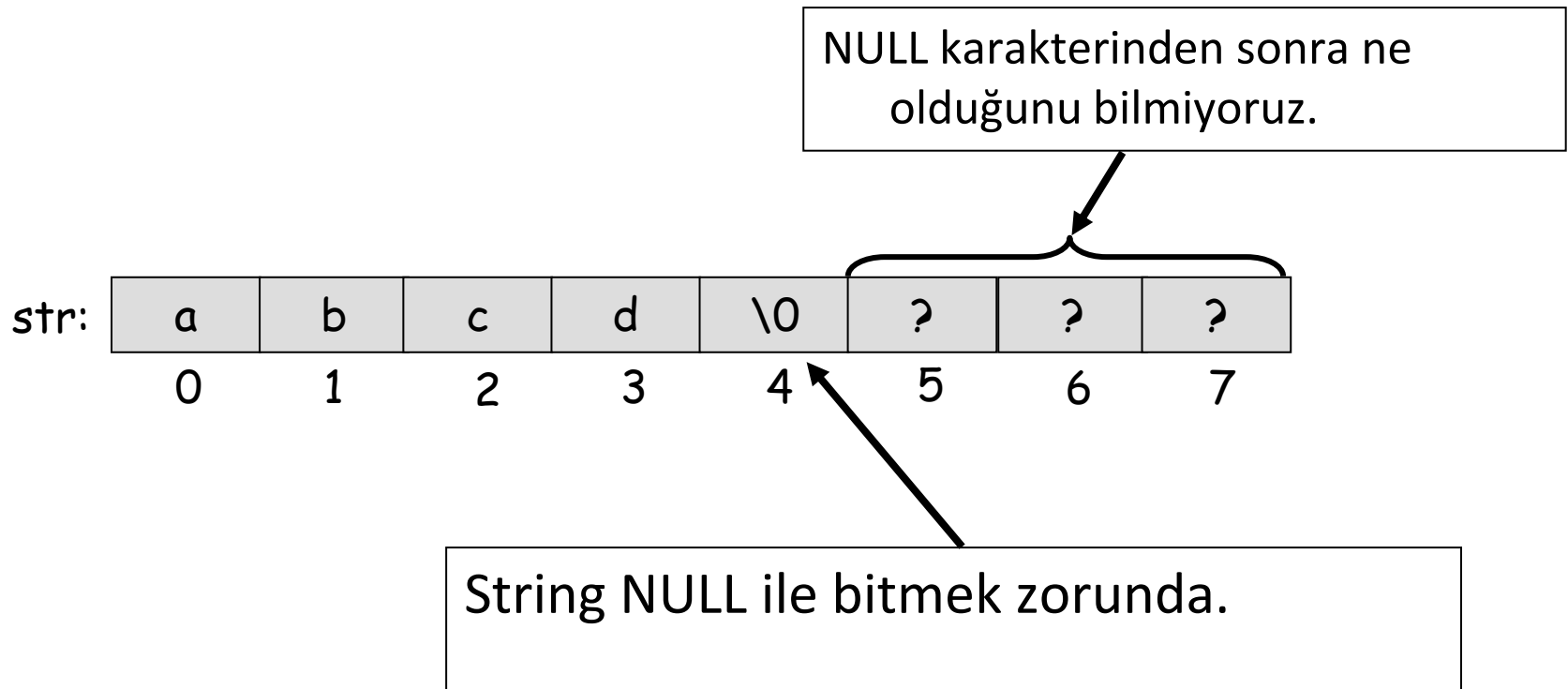
- ▶ Dizileri ve çok boyutlu dizileri gördük.
- ▶ Katar dediğimiz şey de aslında bir dizidir.
- ▶ Değişken tipi char yani karakter olan diziler, 'katar' ya da İngilizce adıyla 'string' olarak isimlendirilirler.
- ▶ Örneğin bir tam sayı (int) dizisinde, tam sayıları saklarken; bir karakter dizisinde -yani katar- karakterleri (char) saklarız.
- ▶ İsimler, adresler, kullanıcı adları, telefonlar vs... sözle ifade edilebilecek her şey için karakter dizilerini kullanırız.

String Tanımlama

- ▶ String NULL karakter `'\0'` ile biten bir karakter dizisidir.
- ▶ Örnek: `char str[8];`
 - En çok 8 karakter alabilen bir dizi oluşturur.
 - Eğer str dizisi string olarak kullanılacak ise en fazla 7 karakter alabilir ve sonu NULL karakter `'\0'` ile bitmek zorunda.

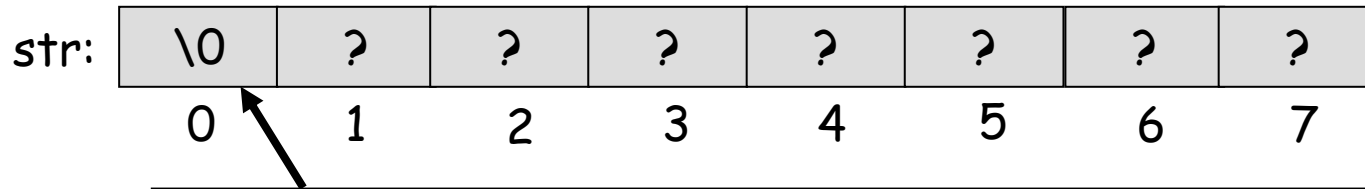
String Gösterimi

- Eğer str de “abcd” yazısını depolarsak bu aşağıdaki şekilde görünecektir.



Boş String

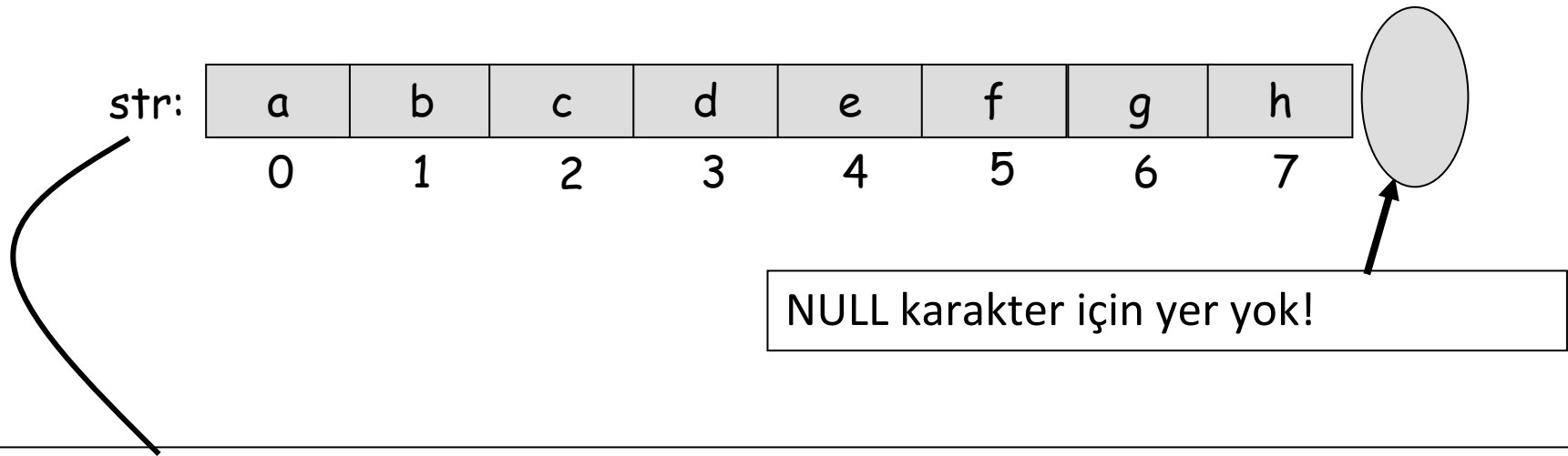
- ▶ Boş string "" ilk elemanı NULL karakter '\0' olan bir karakter dizisini ifade eder.



Boş string in ilk karakteri NULL karakteri olur.

String Maksimum Uzunluğu

- ▶ 8 karakter uzunluğunda bir string örneğin, “abcdefgh” str de depolanamaz



- Bu 8 karakter içeren bir karakter dizisidir.
- Fakat string DEĞİL. Bir string her zaman NULL karakter ile bitmek ZORUNDA!

String: UYARI

- ▶ Sadece tekrar, `char str[8]` gibi bir beyan basitçe `str` de en fazla 8 karakter saklayabileceğimizi vurgular.
- ▶ Programın çalışması sırasında herhangi bir noktada `str` de 8 den daha fazla karakter saklamak isteyebiliriz.
- ▶ Fakat eğer “`str`” bir string depoluyorsa en fazla $8-1=7$ karakter saklayabiliriz, ve her zaman NULL ile bitmek zorunda.

String: İlk Değer Atama

- Bir karakter dizisi tanımlama anında diğer dizilerde olduğu gibi aşağıdaki gibi tanımlanabilir.

```
char str[8] = { 'a' , 'b' , 'c' } ;
```

- Hatırlayın belirtilmeyen elemanlar ' \0 ' ile dolduruluyordu, ki bu da NULL karakter oluyor.
 - Bu nedenle yukarıdaki beyan aşağıdaki string e karşılık geliyor.

str:

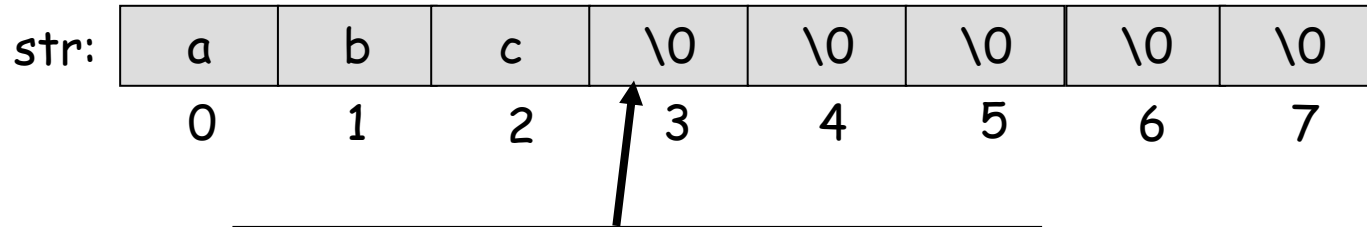
a	b	c	\0	\0	\0	\0	\0
0	1	2	3	4	5	6	7

Gerektiği gibi NULL ile bitiyor.

String: İlk Değer Atama

- ▶ Eğer bir karakter dizisi bir string depolayacaksa, aşağıdaki gibi basitçe ilk değer atanabilir.
 - Sadece string çift tırnak içine konulur. Buna string literalı denir.

```
char str[8] = "abc"; /* önceki ile aynı*/
```



Gerektiği gibi NULL ile bitiyor.

String: İlk Değer Atama

- ▶ Eğer dizinin uzunluğunu tanımlama anında belirtmiyorsa derleyici string uzunluğu + NULL karakter kadar yer ayırır.

```
char str[] = "abc";
```

str:	a	b	c	\0
	0	1	2	3

String: İlk Değer Atama

- ▶ String ler genellikle aşağıdaki gibi tanımlanır.

```
char *str = "abc";
```

str:

a	b	c	\0
0	1	2	3

- ▶ Bu tanımlama ile önceki tanımlamalar arasındaki fark: **bu yol ile tanımlanan string ler READ-ONLY oluyor ve değiştirilemiyorlar.**
- ▶ `char str[]="abc";` şeklinde tanımlanan string leri istediğiniz gibi değiştirebilirsiniz.

String Yazdırmak

- C string leri yazdırmak için iki fonksiyon sunuyor.
 - (1) `puts(str);` (2) `printf(“%s”, str);`

```
char str1[]="bu benim ilk stringim";

/* stringi yazdırır ve imleç sonraki satırın başına geçer.*/
puts(str1);

/* stringi imlecin olduğu yerden yazdırmaya başlar */
printf("%s", str1);

/* 40 boşluk ayırır ve stringi bunun içinde sağa dayalı
   olarak yazdırır. */
printf("%40s", str1);

/* 40 boşluk ayırır ve stringi bunun içinde sola dayalı
   olarak yazdırır. */
printf("%-40s", str1);
```

String Yazdırmak

```
char str1[]="bu benim ilk stringim";

/* stringden sadece ilk 10 karakteri yazar,
 * sağa dayalı */
printf("%.10s", str1);

/* 40 boşluk ayırır ve sadece ilk 10 karakteri yazdırır,
 * sağa dayalı */
printf("%40.10s", str1);

/* 40 boşluk ayırır ve sadece ilk 10 karakteri yazdırır,
 * sola dayalı */
printf("%-40.10s", str1);
```

String Okumak

- ▶ Klavyeden string almak için C iki fonksiyon sunuyor.
 - (1) `gets(str);` (2) `scanf("%s", str);`

```
char str2[80];  
  
/* '\n' girilene kadar girilen stringi okur. */  
gets(str2);  
  
/* bütün boşluk karakterlerini (space, tab, newline) geçerek  
   * girileni sonraki boşluk karakterine kadar okur.*/  
scanf("%s", str2);
```

String Okumak

```
char str2[80];

/* bütün boşluk karakterlerini (space, tab, newline) geçerek
   * girileni sonraki boşluk karakterine kadar okur.*/
scanf("%s", str2);

/*eğer giriş aşağıdaki gibi ise:_ space olarak farz edelim */
_ _xyz123_ _ _45_ _67
```

- ▶ scanf ilk iki boşluğu geçecek ve str2 “xyz123” olacak.
- ▶ Sonra boşluğu görecektir ve okuma duracaktır.
- ▶ Bir sonraki scanf(“%s”, ...) bu boşlukları geçecek ve “45” i okuyacaktır.

String İşlemleri

- ▶ C standard kütüphanesi string leri manipüle etmek için birçok fonksiyon içeriyor.
 - Bu fonksiyonları kullanmak için `<string.h>` dosyasını eklemeniz gerekiyor. `#include <string.h>`
- ▶ Bazı önemli fonksiyonlar:
 - `strlen(const char *str);`
 - `strcpy(char *str1, const char *str2);`
 - `strcat(char *str1, const char *str2);`
 - `strcmp(const char *str1, const char *str2);`
- ▶ Bu fonksiyonların detaylarına önümüzdeki dönem gireceğiz.

Örnek: Karakter Dizisinin Uzunluğu

```
#include <stdio.h>

int main(void){
    char s[40];
    int k = 0;

    /* diziye oku */
    printf("Bir şeyler yazın : ");
    gets(s);

    /* sonlandırıcı karaktere kadar karakterleri say */
    while( s[k]!='\0' )
        k++;
    printf("Dizinin uzunluğu : %d\n",k);

    return 0;
}
```

Örnek: Karakter Dizisinin Tersini

```
#include <stdio.h>

int main(void){
    char s[40], gecici;
    int i, n;
    /* diziyi oku */
    printf("Bir seyler yazin : ");
    gets(s);
    /* sonlandırıcı karaktere kadar */
    for(n=0; s[n] != '\0'; n++);
    for(i=0; i<n/2; i++){
        gecici = s[n-i-1];
        s[n-i-1] = s[i];
        s[i] = gecici;
    }
    printf("Tersi      : %s\n",s);
    return 0;
}
```

Kaynaklar

- ▶ Doç. Dr. Fahri Vatansever, “Algoritma Geliştirme ve Programlamaya Giriş”, Seçkin Yayıncılık, 12. Baskı, 2015.
- ▶ J. G. Brookshear, “Computer Science: An Overview 10th Ed.”, Addison Wisley, 2009.
- ▶ Kaan Aslan, “A’dan Z’ye C Klavuzu 8. Basım”, Pusula Yayıncılık, 2002.
- ▶ Paul J. Deitel, “C How to Program”, Harvey Deitel.
- ▶ Bayram AKGÜL, C Programlama Ders notları