

ALGORİTMALAR VE PROGRAMLAMA II

Hafta 5

String Fonksiyonlar

String

String NULL karakter '\0' ile biten bir karakter dizisidir.

Örnek: `char str[8];`

- ▶ Ençok 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 zorundadır.



String İşlemleri

- ▶ C standard kütüphanesi stringleri manipüle etmek için birçok fonksiyon içerir.
- ▶ Bu fonksiyonları kullanmak için `<string.h>` ifadesini eklemeniz gerekir.

#include <string.h>

- ▶ Bazı önemli fonksiyonlar:
 - **strcpy(char *str1, const char *str2);**
 - **strlen(const char *str);**
 - **strcat(char *str1, const char *str2);**
 - **strcmp(const char *str1, const char *str2);**

Strcpy Fonksiyonu

```
strcpy(char *str1, const char *str2)
```

- `strcpy`, `str2` string'ini `str1` 'e kopyalar. `str1` 'in yeterli belleğe sahip olması gerekir.

```
char *strcpy(char *str1, const char *str2)
{
    char *p = str1;

    while (*str2)
        *p++ = *str2++;

    *p = '\0';
    return str1;
} /* end-strcpy */
```



Strcpy Fonksiyonu

```
#include <stdio.h>
#include <string.h>
int main () {
    char src[40];
    char dest[100];
    strcpy(src, "This is
tutorialspoint.com"); strcpy(dest, src);
    printf("Final copied string : %s\n", dest);
    return(0);
}
```

Output:

Final copied string : This is
tutorialspoint.com



Strncpy Fonksiyonu

```
strncpy(char *str1, const char *str2, size_t n)
```

- `str2` string'inin ilk `n` karakterini `str1` 'e kopyalar.

```
#include <stdio.h>#include <string.h>
int main()
{   char src[] = "Merhaba Dünya";
    char dest[20];
    strncpy(dest, src, 7);
    dest[7] = '\0'; //Null karakter eklemek
    printf("Kopyalanan string: %s\n", dest);
    return 0;
}
```

• Strlen Fonksiyonu

`strlen(const char *str)`

- `String`'in uzunluğunu döner (`\0` hariç).

```
int  strlen(const char
*str)
{
    int len = 0;

    while(*str++)
        len++;

    return len;
} /* end-strlen */
```



Strlen Fonksiyonu

```
#include <stdio.h>
#include <string.h>
int main () {
    char str[50];
    int len;
    strcpy(str, "This is
tutorialspoint.com"); len = strlen(str);
    printf("Length of |%s| is |%d|\n", str,
len);
    return(0);
}
```

Output:

Length of |This is tutorialspoint.com| is |26|



Strcat Fonksiyonu

```
strcat(char *str1, const char *str2)
```

- `str2` string'ini `str1` 'in sonuna ekler.

```
char *strcat(char *str1, const char *str2)
{
    char *p = str1;
    while(*p)
        p++;
    while(*str2)
        *p++ = *str2++;
    *p = '\\0';
    return str1;
} /* end-strcat */
```



• Strcat Fonksiyonu

```
#include <stdio.h>
#include <string.h>
int main () {
    char src[50],
    dest[50];
    strcpy(src, "This is source");
    strcpy(dest, "This is
destination"); strcat(dest, src);
    printf("Final destination string : |%s|",
    dest); return(0);
}
```

Output:

Final destination string : |This is destinationThis is
source|

Strncat Fonksiyonu

```
strncat(char *str1, const char *str2, size_t n)
```

- `str2` string'inin ilk `n` karakterini `str1`'in sonuna ekler.

```
#include <stdio.h>
#include <string.h>
int main()
{
    char str1[30] = "Merhaba ";
        char str2[] = "Dünya";
        strncat(str1, str2, 3);
    printf("Sonuç: %s\n", str1);
    return 0;
}
```



Strcmp Fonksiyonu

```
strcmp(const char *str1, const char *str2)
```

- İki string'i karşılaştırır.

```
int    strcmp(const char *str1, const char
*str2)
{
    while (*str1 && *str2 && *str1 == *str2) {
        str1++; str2++;
    } /* end-while */

    return *str1-*str2;
} /* end-strcmp */
```



Strcmp Fonksiyonu

```
#include <stdio.h>
#include <string.h>
int main () {
    char str1[15], str2[15];
    int ret;
    strcpy(str1, "abcdef");
    strcpy(str2, "ABCDEF");
    ret = strcmp(str1,
str2); if(ret < 0)
        printf("str1 is less than
str2"); else if(ret > 0)
        printf("str2 is less than str1");
    else
        printf("str1 is equal to str2");
    return(0);
}
```

Output:

str2 is less than str1

A decorative horizontal bar at the bottom of the slide, composed of several colored rectangular segments in shades of blue, green, yellow, orange, red, and grey.

Strncmp Fonksiyonu

```
strncmp(const char *str1, const char *str2, size_t n)
```

- İlk `n` karakteri karşılaştırır.

```
#include <stdio.h>
#include <string.h>
int main()
{   char str1[] = "Merhaba Dünya";
    char str2[] = "Merhaba Mars";
    if (strncmp(str1, str2, 7) == 0)
    {   printf("İlk 7 karakter eşittir.\n");
    } else
    {   printf("Farklıdır.\n");
    }
    return 0;
}
```



Strcasecmp Fonksiyonu

```
strcasecmp(const char *str1, const char *str2)
```

- Harf duyarsız karşılaştırma (Unix sistemlerinde).

```
#include <stdio.h>
#include <strings.h> // strcmp için gerekli
int main() {
    const char *str1 = "Merhaba";
    const char *str2 = "MERHABA";
    int result = strcmp(str1, str2);
    if (result == 0) {
        printf("İki string eşittir.\n");
    } else if (result < 0) {
        printf("str1, str2'den alfabetik olarak önce gelir.\n");
    } else {
        printf("str1, str2'den alfabetik olarak sonra gelir.\n");
    }
    return 0;
}
```

Strchr Fonksiyonu

```
strchr(const char *str, int c)
```

- Belirli bir karakteri arar.

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

```
#include <string.h>
```

```
int main()
```

```
{    char str[] = "Merhaba Dünya";
```

```
    char *pos = strchr(str, 'D');
```

```
    if (pos != NULL)
```

```
    {
```

```
        printf("Karakter bulundu: %s\n", pos);
```

```
    }
```

```
    return 0;
```

```
}
```


Strchr Fonksiyonu

- string içinde aranan karakterin **son geçtiği yerin adresini** döner.
- Eğer karakter bulunamazsa NULL döner.
- Son geçtiği yerden itibaren string'in geri kalan kısmını gösterir.
- Bu fonksiyon, genellikle dosya uzantısı bulma gibi işlemlerde çok kullanışlıdır.

```
strchr(const char *str, int c)
```

- Son geçen karakteri arar.

• Strchr Fonksiyonu

```
#include <stdio.h>
#include <string.h> // strchr için gerekli
int main() {
    const char *str = "Merhaba dünya!";
    char ch = 'a';
    char *result = strchr(str, ch);
    if (result != NULL)
    {
        printf("'%'c' karakterinin son geçtiği yer: %s\n", ch,
        result);
    } else {
        printf("Karakter bulunamadı.\n");
    }
    return 0;
}
```

• Strstr Fonksiyonu

- bir string içinde **alt string'i arar** ve bulunduğu yerin adresini döner.
- Eğer alt string bulunamazsa NULL döner.
- Bulunan kısımdan itibaren string'in geri kalanını gösterir.
- Bu fonksiyon, metin arama işlemlerinde çok kullanışlıdır.



• Strstr Fonksiyonu

```
#include <stdio.h>
#include <string.h> // strstr için gerekli
int main() {
    const char *text = "C programlama dilini
öğreniyorum.";
    const char *search = "dil";
    char *result = strstr(text, search);
    if (result != NULL) {
        printf("'%' ifadesi bulundu: %s\n", search,
result);
    } else {
        printf("İfade bulunamadı.\n");
    }
    return 0;
}
```



Strspn Fonksiyonu

```
strspn(const char *str1, const char *str2)
```

- `str1` 'in başında `str2` karakterlerinden oluşan kısmın uzunluğunu döner.
- `string`'in **başından itibaren**, belirtilen karakter kümesine **ait olan karakterleri sayar**.
- İlk uygun olmayan karaktere gelince durur.

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

```
#include <string.h> // strspn için gerekli
```

```
int main() {
```

```
    const char *str = "12345abc678";
```

```
    const char *accept = "1234567890";
```

```
    size_t result = strspn(str, accept);
```

```
    printf("Sadece rakamlardan oluşan kısmın  
uzunluğu: %zu\n", result);
```

```
    return 0;
```

```
}
```



Atoi Fonksiyonu

- **ASCII to Integer** anlamına gelir.
- String içinde bulunan sayısal karakterleri tamsayıya çevirir.
- Eğer string sayısal olmayan karakter içeriyorsa, ilk sayısal olmayan karaktere kadar olan kısmı dönüştürür.

```
#include <stdio.h>
#include <stdlib.h> // atoi için gerekli
int main() {
    const char *str = "12345";
    int number;
    number = atoi(str);
    printf("String olarak: %s\n", str);
    printf("Tamsayı olarak: %d\n", number);
    return 0;
}
```



• Atof Fonksiyonu

- **ASCII to Float** anlamına gelir.
- String içindeki **ondalıklı sayıyı, double türüne dönüştürür.**
- Eğer string, sayısal olmayan karakter içeriyorsa, ilk geçerli kısmı alır.
- **hatalı girişlerde 0 döner ve hata kontrolü yapmaz.**

```
#include <stdio.h>
#include <stdlib.h> // atof için gerekli
int main() {
    const char *str = "3.14159";
    double number;
    number = atof(str);
    printf("String olarak: %s\n", str);
    printf("Ondalıklı sayı olarak: %f\n", number);
    return 0;
}
```



• Strtol Fonksiyonu

- string içindeki **geçerli sayı kısmını dönüştürür** ve **geçersiz kısmın adresini endPtr üzerinden döner**.

```
#include <stdio.h>
#include <stdlib.h> // strtol için gerekli
int main() {
    const char *str = "1234abc";
    char *endPtr;
    long number;
    number = strtol(str, &endPtr, 10); // 10 tabanında
    çevir
    printf("Çevrilen sayı: %ld\n", number);
    if (*endPtr != '\0') {
        printf("Geçersiz kısım: %s\n", endPtr);
    }
    return 0;
}
```



• Strtod Fonksiyonu

- **ASCII to Double** anlamına gelir.
- **Ondalıklı sayının geçerli kısmını çevirir ve geçersiz kısmı endPtr üzerinden döner.**


```
#include <stdio.h>
#include <stdlib.h> // strtod için gerekli
int main() {
    const char *str = "3.14159abc";
    char *endPtr;
    double number;
    number = strtod(str, &endPtr);
    printf("Çevrilen sayı: %lf\n", number);
    if (*endPtr != '\0') {
        printf("Geçersiz kısım: %s\n", endPtr);
    }
    return 0;
}
```



Doç. Dr. Caner ÖZCAN, KBÜ Yazılım Mühendisliği www.canerozcan.net*

- ▶ Doç. Dr. Fahri Vatansever, “Algoritma Geliştirme ve Programlamaya Giriş”, Seçkin Yayıncılık, 12. Baskı, 2015.
- ▶ Kaan Aslan, “A’dan Z’ye C Klavuzu 8. Basım”, Pusula Yayıncılık, 2002.
- ▶ Paul J. Deitel, “C How to Program”, Harvey Deitel.
- ▶ “A book on C”, All Kelley, İra Pohl

* Bu dersin slaytları genelde bu kaynaktan türetilmiştir.



Ders içerikleri ve duyurular için dersin web sitesine
aşağıdaki adresten ulaşabilirsiniz.

hru-algpro.github.io

S o r u l a r

?



Dinlediğiniz için teşekkürler

