

İÇİNDEKİLER

▪ İÇİNDEKİLER	1
▪ AÇIKLAMA	2
▪ FT_STRLEN()	3
▪ FT_STRCPY()	4
▪ FT_STRNCPY()	5
▪ FT_STRCMP()	6
▪ FT_STRNCMP()	7
▪ FT_STRCAT()	8
▪ FT_STRNCAT()	9
▪ FT_STRSTR()	10

AÇIKLAMA

> Merhaba ben Can,

Bu PDF dökümantasyonunu oluştururken amacım, C dilindeki string fonksiyonlarını anlaşılabilir bir şekilde açıklamak ve C programlama dilini öğrenmek isteyen herkese yardımcı olmaktır. Ticari bir amacı kesinlikle yoktur, tamamen eğitim ve öğrenme amacı taşımaktadır.

C programlama dili, yazılım dünyasının temel taşlarından biri olarak kabul edilir ve çeşitli uygulamalarda sıkça kullanılır. Ancak, dilin bazı özellikleri öğrenenler için başlangıçta karmaşık gelebilir. Bu dökümantasyon, özellikle C dilindeki string işlemleri üzerine odaklanarak, bu karmaşıklığı azaltmaya ve kullanıcıların dilin temel fonksiyonlarını anlamasına yardımcı olmayı amaçlamaktadır.

Her bir string fonksiyonunu, basit ve anlaşılır örneklerle birlikte adım adım açıklamalarla sunmaya çalıştım. Bu sayede, herkesin C dilinde string fonksiyonlarını daha iyi anlamasını hedefledim. Ayrıca, kod örneklerinin pratik uygulamalarını anlatarak gerçek dünya senaryolarına dair bir bakış sunmaya çalıştım.

Eğer bu dökümantasyondan faydalanan ve daha fazla içerik görmek isteyenler olursa, [c4nn.g](#) Instagram hesabımı ve GitHub profilimi takip ederek güncel bilgiler ve yeni kaynaklarla daha da zenginleşen içeriklere ulaşabilirler.

Instagram: [@c4nn.g](#) GitHub: [@c4nng](#)

Unutmayın, amacımız C programlama dilini daha iyi anlamak ve öğrenmek. Bu süreçte her türlü sorunuzu sormaktan çekinmeyin, dilin güzelliklerini keşfetmek için beraber çalışabiliriz.

C dilindeki string fonksiyonları ve daha fazlası için bu dökümantasyonu takip ederek ilerleyebilirsiniz. Umarım bu döküman, dilinizi geliştirmenize ve yazılım dünyasında daha güvenle yol almanıza yardımcı olur.

Saygılarımla, CAN (r00t)

>> FT_STRLEN()

strlen() fonksiyonu, verilen bir karakter dizisinin (string) uzunluğunu döndürür. Yani, kaç karakterden oluştuğunu hesaplar.

<Örnek Kullanım>

```
ft_strlen.c

1 #include "stdio.h"
2
3 int ft_strlen(char *str)
4 {
5     int i;
6     i = 0;
7
8     while(str[i] != '\0')
9     {
10         i++;
11     }
12     return i;
13 }
14
15 int main()
16 {
17     char g[] = "Enes";
18     int result;
19
20     result = ft_strlen(g);
21     printf("STRLEN | %d",result);
22 }
```

>> FT_STRCPY()

strcpy() fonksiyonu, bir kaynak karakter dizisini hedef karakter dizisine kopyalar.

<Örnek Kullanım>

```
ft_strcpy.c

1 #include "stdio.h"
2
3 char *ft_strcpy(char *dest, char *src)
4 {
5     int i;
6     i = 0;
7
8     while (src[i] != '\0')
9     {
10         dest[i] = src[i];
11         i++;
12     }
13     dest[i] = '\0';
14     return dest;
15 }
16
17 int main()
18 {
19     char src[] = "Can";
20     char dest[] = "Enes";
21     char *result;
22     result = ft_strcpy(dest,src);
23     printf("%s",result);
24 }
```

>> FT_STRNCPY()

strncpy() fonksiyonu, belirtilen sayıda karakteri bir kaynak karakter dizisinden hedef karakter dizisine kopyalar. Eğer kaynak dizi, belirtilen sayıda karakterden daha kısa ise hedef dizi sonlandırıcı null karakterlerle doldurulur.

<Örnek Kullanım>

```
ft_strcpy.c

1 #include "stdio.h"
2
3 char *ft_strncpy(char *dest, char *src, unsigned int n)
4 {
5     unsigned int i;
6     i = 0;
7
8     while (src[i] != '\0' && i < n)
9     {
10         dest[i] = src[i];
11         i++;
12     }
13
14     while (i < n)
15     {
16         dest[i] = '\0';
17         i++;
18     }
19     return (dest);
20 }
21
22 int main()
23 {
24     char src[] = "Gonca";
25     char dest[] = "Enes";
26     char *result;
27     result = ft_strncpy(dest,src,6);
28     printf("%s",result);
29 }
```

>> FT_STRCMP()

strcmp() fonksiyonu, iki karakter dizisini karşılaştırarak leksikografik olarak (sözlük sırasına göre) hangisinin önce geldiğini, eşitse, daha büyük veya daha küçük olduğunu belirler.

<Örnek Kullanım>

```
ft_strcmp.c

1 #include "stdio.h"
2
3 int ft_strcmp(char *s1, char *s2)
4 {
5     int i;
6     i = 0;
7
8     while(s1[i] != '\0' || s2[i] != '\0')
9     {
10         if(s1[i] > s2[i])
11         {
12             return 1;
13         }
14
15         else if(s1[i] < s2[i])
16         {
17             return -1;
18         }
19         i++;
20     }
21     return 0;
22 }
23
24 int main()
25 {
26     char g[] = "Gonca";
27     char e[] = "Gonca";
28
29     int result;
30
31     result = ft_strcmp(g,e);
32     printf("Result | %d",result);
33 }
```

>> FT_STRNCMP()

strncmp() fonksiyonu, belirtilen iki karakter dizisini, belirtilen maksimum uzunluğa kadar karşılaştırır. strcmp'ten farklı olarak, karakter sayısı belirterek karşılaştırma yapmayı sağlar.

<Örnek Kullanım>

```
ft_strncmp.c

1 #include "stdio.h"
2
3 int ft_strncmp(char *s1, char *s2, unsigned int n)
4 {
5     int i;
6     i = 0;
7
8     while(s1[i] != '\0' || s2[i] != '\0' && i < n)
9     {
10         if(s1[i] > s2[i])
11         {
12             return 1;
13         }
14         else if(s1[i] < s2[i])
15         {
16             return -1;
17         }
18         i++;
19     }
20     return 0;
21 }
22
23 int main()
24 {
25     char g[] = "enes";
26     char e[] = "gggg";
27
28     int result;
29     result = ft_strncmp(g,e,3);
30
31     printf("Result | %d",result);
32 }
```

>> FT_STRCAT()

strcat() fonksiyonu, iki karakter dizisini birleştirerek bir hedef karakter dizisine ekler.

<Örnek Kullanım>

```
ft_strcat.c

1 #include "stdio.h"
2
3 char *ft_strcat(char *dest, char *src)
4 {
5     int i = 0;
6     int j = 0;
7
8     while (dest[i] != '\0')
9     {
10         i++;
11     }
12
13     while (src[j] != '\0')
14     {
15         dest[i] = src[j];
16         i++;
17         j++;
18     }
19     dest[i] = '\0';
20     return dest;
21
22 }
23
24
25 int main()
26 {
27     char dest[10] = "Enes";
28     char src[] = " Can";
29
30     char *result;
31     result = ft_strcat(dest, src);
32
33     printf("Result | %s", result);
34 }
```


>> FT_STRNCAT()

strncat() fonksiyonu, belirtilen iki karakter dizisini, belirtilen maksimum uzunluğa kadar hedef karakter dizisine ekler.

<Örnek Kullanım>

```
ft_strncat.c

1 #include "stdio.h"
2
3 char *ft_strncat(char *dest, char *src, unsigned int nb)
4 {
5     int i = 0;
6     int j = 0;
7
8     while(dest[i] != '\0')
9     {
10         i++;
11     }
12
13     while(src[j] != '\0' && j < nb)
14     {
15         dest[i] = src[j];
16         i++;
17         j++;
18     }
19     dest[i] = '\0';
20     return dest;
21 }
22
23 int main()
24 {
25     char dest[20] = "Enes";
26     char src[] = " Cannn";
27
28     char *result;
29     result = ft_strncat(dest, src, 4);
30
31     printf("Result | %s", result);
32 }
```

>> FT_STRSTR()

Açıklama: strstr() fonksiyonu, bir karakter dizisi içinde belirtilen bir alt dizeyi arar ve ilk bulunduğu konumun adresini döndürür.

<Örnek Kullanım>

```
ft_strstr.c

1 #include "stdio.h"
2 char *ft_strstr(char *str, char *to_find)
3 {
4     while (*str != '\0')
5     {
6         const char *s1 = str;
7         const char *s2 = to_find;
8
9         while (*s1 == *s2 && *s2 != '\0')
10        {
11            s1++;
12            s2++;
13        }
14        if (*s2 == '\0')
15        {
16            return (char *) str;
17        }
18        str++;
19    }
20 }
21
22 int main()
23 {
24     char str[] = "Enes Can 1235";
25     char to_find[] = "Can";
26     char *result;
27     result = ft_strstr(str, to_find);
28     printf("%s", result);
29 }
```