

Ders 31

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

int main(void)
{
    char str[1000];

    //code
    size_t len = strlen(str);
}
```

Tür eş isimleri fonksiyonların geri dönüş değeri olarak taşımayı kolaylaştırır.

Örnek

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

time_t time(time_t* p);

int main(void)
{
    long long second;

    time(&second);
}
```

Burada derleyici hata vermese bile başka bir derleyicide bu fonksiyon farklı türden tanımlanmış olabilir ve tanımsız davranış olabilir bu yüzden tür eş isimleriyle kullanırız.

Time_t second gibi

En önemlisi size_t

Örnek p1-p2

```

#include <stdio.h>
#include <string.h>
#include <time.h>
#include <stdlib.h>

int main(void)
{
    int a[100];

    int* p1 = a + 5;
    int* p2 = a + 34;
    //

    p1 - p2
}

```

İki adres birbirinden çıkartılabilir negatif değerde olabilir bunun türü ptrdiff_t dir.

```

#include <stdio.h>
#include <string.h>
#include <time.h>
#include <stdlib.h>

int main(void)
{
    double* p;
    //...

    intptr_t val = (intptr_t)p;
}

```

Bir pointerin değerinin tam sayılarda tutulması.

STANDART STRING KÜTÜPHANELERİ.

Örnek fgets

```
#define _CRT_SECURE_NO_WARNINGS
```

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

```
int main(void)  
{  
    char str[10];  
  
    printf("bir yazı girin: ");  
    sgets(str);  
  
    puts(str);  
}
```

Diziye 9 karakterden fazlasını girersem dizi taşar ve tanımsız davranış olur.

Önemli string fonksiyonlar

<string.h>

=====

strlen

strchr

strrchr

strstr

strpbrk

strcpy

strcat

strspn

strcspn

strcmp

strncpy

strncat

strncmp

strtok

strerror

Örnek kende puts fonksiyonumuzu yazmak

```
#define _CRT_SECURE_NO_WARNINGS
```

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

```
void myputs(const char *p)  
{  
    while (*p != '\0')  
        putchar(*p++);  
  
    putchar('\n');  
}
```

```
int main(void)  
{  
  
}
```

orjinalinden tek farkı geri dönüş değeri

```
int main(void)  
{  
    char str[] = "ılayda toprak";  
    myputs(str);  
}
```

ılayda toprak yazdı.

Aynı örnekte aynı giriş ile sadece soyad toprak nasıl yazılırdı.

Fonksiyona giriş argümanı (str+7) olsaydı t harfinden başlardı yazmaya.

Örnek

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

```
int main(void)  
{  
    char str[] = "ılayda toprak";  
  
    for (int i = 0; i < 13; ++i)  
        puts(str + i);  
}
```

Çıktısı.

```
Microsoft Visual Studio Debug Console
ilayda toprak
layda toprak
ayda toprak
yda toprak
da toprak
a toprak
 toprak
toprak
oprak
prak
rak
ak
k
```

Aslında anlamamız gereken fonksiyona adresi neresi verirsek sonuna kadar yazdırıyor.

Sonunda null olduğuna güvenerek bu işlemleri yapıyor.

Bu adres verme biçimini normal fonksiyonlardada kullanabilirim.

Örnek strlen

```
#define _CRT_SECURE_NO_WARNINGS

#include "nutility.h"
#include <stdio.h>
#include <string.h>

#define SIZE 1000

int main(void)
{
    char str[SIZE];

    printf("bir yazi girin: ");
    scanf("%s", str);
    size_t len = strlen(str);

    printf("uzunluk = %zu\n", len);
    //...
}
```

Yazı dizisinin boyutunu hesaplıyoruz.

Scanf içinde & yok çünkü str zaten ilk adresi array decay

Strlen i kendimiz yazalım.

```
size_t mystrlen(const char* p)
{
    size_t len = 0;

    while (*p++ != '\0') {
        ++len;
    }

    return len;
}
```

```
size_t mystrlen(const char* p)
{
    size_t len = 0;

    while (*p++) {
        ++len;
    }

    return len;
}
```

*p++ null yani 0 olduğunda döngüden çıkacak

```
size_t mystrlen(const char* p)
{
    const char* ptemp = p;

    while (*p)
        ++p;

    return (size_t)(p - ptemp);
}
```

En sık kullanılan fonksiyon son yazdığımız adresten adres çıkartıp boyutu bulmak.

Örnek yazıyı ters çeviren kod

```

void rputs(const char* p)
{
    for (int i = (int)strlen(p) - 1; i >= 0; --i) {
        putchar(p[i]);
    }
    putchar('\n');
}

int main(void)
{
    char str[SIZE];

    printf("bir yazi girin: ");
    scanf("%s", str);

    puts(str);
    rputs(str);
}

```

Strlen fonksiyonunu fonksiyon içinde araç olarak kullandık.

Örnek strchr

```

#define      SIZE      100

int main(void)
{
    char str[SIZE];
    int c;

    printf("bir yazi girin: ");
    sgets(str);
    printf("aranacak karakteri girin: ");
    c = getchar();

    // murat gunay      g

    char* p = strchr(str, c);
    if (p) {
        printf("bulundu... (%c) karakteri (%d) indisinde (%s)\n", *p, p - str, p);
    }
    else {
        printf("bulunamadi\n");
    }
}

```

Bu kodda strchr foksiyonda bulduğu yerin adresini döndürüyor.

if(p) ile if(p!=null) aynı ifadedir

Yani printf deki *p adresteki karakter

p-str pointer aritmetiğinden dolayı indissayısı

%s str de adresten itibaren null a kadar olan yazıyı yazar.

```
int main(void)
{
    char str[SIZE] = "kerem zorer";

    for (int i = 0; i < (int)strlen(str); ++i) {
        printf("%s\n", str + i);
    }
}
```

Microsoft Visual Studio Debug Console

```
kerem zorer
erem zorer
rem zorer
em zorer
m zorer
 zorer
zorer
orer
rer
er
r
```

Yazı uzunluğu bulma


```

#include "nutility.h"
#include <stdio.h>
#include <string.h>

#define SIZE 100

int main(void)
{
    char str[SIZE] = "kerem zorer";

    char* p = strchr(str, '\0');
    printf("%d\n", p - str);
}

```

Devam yazının sonuna karakter ekleme

```

#include "nutility.h"
#include <stdio.h>
#include <string.h>

#define SIZE 100

int main(void)
{
    char str[SIZE] = "kerem zorer";

    char* p = strchr(str, '\0');
    *p++ = '!';
    *p = '\0';

    printf("(%s)\n", str);
}

```

Microsoft Visual Studio Debug Console

```
(kerem zorer!)
```

Örnek c idiomu

NULL	'\0'
bir makro	character constant
pointer	tam sayı
(void *)	int
ptr = NULL	str[]

öyle bir C idiyomu var ki

```
char *ptr;
```

```
///
```

p bir yazıyı gösteriyor ise (yani değeri NULL pointer değil ise) ve ptr'nin gösterdiği yazı boş değil ise I

```
if (p != NULL && *p != '\0')
```

p bir yazıyı gösteriyor ise (yani değeri NULL pointer değil ise) ve p'nin gösterdiği yazı boş değil ise

```
if (p != NULL && *p != '\0')
if (p && *p)
```

p bir yazıyı göstermiyor ise (yani değeri NULL pointer ise) veya p'nin gösterdiği yazı boş ise I

```
if (p == NULL || *p == '\0')
if (!p || !*p)
```

////////////////////////////////////

Mülakat Soruların cevabı nedir.

```
#include "nutility.h"
#include <stdio.h>
#include <string.h>

#define SIZE 100

int main(void)
{
    char str[100] = "MURAT";
    char* p = str;

    printf("[1] %zu\n", sizeof str);
    printf("[2] %zu\n", sizeof *str);
    printf("[3] %zu\n", sizeof p);
    printf("[4] %zu\n", sizeof *p);
    printf("[5] %zu\n", sizeof p++);
    printf("[6] %zu\n", strlen(p));
}
```

Benim cevabım. Hocanın cevabı

1 100

2. 1

3. 100 4

4. 1

5. 100 4

6. 5

Zor olanı yaptım kolayda patladım.

Pointer türün boyutu derleyicide 4 byte bu yüzden 4

p türü char oda 1 doğru yaptım..

Burada şaşırtma p++ imiş ama sizeof içinde işleme alınmaz.

Mülakat sorusu 2

```
#define      SIZE      100

int main(void)
{
    printf("%zu\n", sizeof("kenan"));
    printf("%zu\n", strlen("kenan"));
}
I
```

Size of 6

Strlen 5

String literalide bir dizidir.

İdiom

```
#include "nutility.h"
#include <stdio.h>
#include <string.h>

#define SIZE 100

int main(void)
{
    char str[100];

    if (strchr(str, 'a')) { //yazının içinde a karakteri varsa
    }
}
```

!= null yerine yalın

```
#include "nutility.h"
#include <stdio.h>
#include <string.h>

#define SIZE 100

int main(void)
{
    char str[100];

    if (!strchr(str, 'a')) { //yazının içinde a karakteri yoksa
    }
}
```

////////////////////////////////

```
p = strchr(str, 'a');
if (p != NULL) {
}
```

///

```
if ((p = strchr(str, 'a')) != NULL) {
}
```

2 kodda aynı atama operandının ürettiği değerden faydalandık.

////////////////////////////////

Örnek kendi strchr kodu.

```

1
//|
char* mystrchr(const char* p, int c)
{
    while (*p) {
        if (*p == c)
            return p;

        ++p;
    }

    return NULL;
}

```

Burada hatayı görelim diye durdu hata değeri const charp geri dönüş değeri char olarak dönderiyoruz o yüzden tür dönüştürme yapmalıyım.

```

#include <stdio.h>
#include <string.h>

#define SIZE 100

char* mystrchr(const char* p, int c)
{
    while (*p) {
        if (*p == c)
            return (char *)p;

        ++p;
    }

    if (c == '\0')
        return (char*)p;

    return NULL;
}

```

Güzel örnek