

Ders 30

NULL bir makrodur.

bir anahtar sözcük değildir.

Bir identifier değildir.

Bu makro C'nin standart kütüphanesinin birçok başlık dosyasında tanımlanmıştır (define edilmiştir)

```
<stdio.h>
<string.h>
<stddef.h>
<time.h>
<stdlib.h>
```

NULL bir adres sabitidir. Dolayısıyla sadece pointer değişkenlere ilk değer olarak verilebilir ya da atanabilir. pointer olmayan bir değişkene (int, double) asla NULL atamayın.

Değeri NULL pointer olan bir pointer değişken

a) geçerli bir durumdadır

b) ancak hiçbir nesneyi göstermemektedir.

Değeri NULL pointer olan bir pointer değişkeni

a) dereference etmek

yani * ya da [] operandı yapmak

++ptr

--ptr

ptr + n

```
if (ptr == NULL)
```

```
if (ptr != NULL)
```

```
if (p1 == p2)
```

Örnek str

```
#define SIZE 100

int main()
{
    char str[SIZE];

    printf("bir yazi girin : ");
    sgets(str);
    char* p = strchr(str, 'a');
    if (p) {
        printf("bulundu yazinin %d indisli karakteri\n", p - str);
    }
    else {
        printf("bulunamadi\n");
    }
}
```

A karakteri var mı yok mu diye bakıyoruz.

Bulunursa adresi bulunamazsa null pointer.

Örnek kendi arama fonksiyonumuz.

Öncetest kodunu yazdık

```
int main(void)
{
    int a[SIZE];

    randomize();
    set_array_random(a, SIZE);
    print_array(a, SIZE);
    int sval;

    printf("aranan degeri giriniz: ");
    scanf("%d", &sval);

    int* p = search_in_array(a, SIZE, sval);
    if (!p) {
        printf("aranan deger bulunamadi\n");
    }
    else {
        printf("bulundu %d dizinin %d indisli elemani\n", *p, p - a);
    }
}
```

P! P==0 ile aynı anlamlı unutmak

Aranan nesne indisi için p-a yaptı adres aritmetiği.

```
#define SIZE 100

int* search_in_array(const int* p, int size, int key)
{
    while (size-- > 0) {
        if (*p == key)
            return (int*)p;
        ++p;
    }
    return NULL;
}

//int* search_in_array(const int* p, int size, int key)
//{
//    for (int i = 0; i < size; ++i) {
//        if (p[i] == key)
//            return (int*)(p + i);
//    }
//    return NULL;
//}

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

    randomize();
    set_array_random(a, SIZE);
    print_array(a, SIZE);
    int sval;
```

2 türlüde yazabiliriz kodu hatta ben *p++ ilede yazabilirdim.

Dikkat dökümantasyon

dikkat!

adres döndüren fonksiyonlarda aşağıdaki noktaları belgelemeyi asla ihmal etmeyin

1. statik ömürlü bir nesne adresi mi
2. dinamik ömürlü bir nesne adresi mi
3. benim sana gönderdiğim adresi mi bana gönderiyorsun

NULL pointer döndürme ihtimali var mı?

parametresi pointer olan fonksiyonlar

fonksiyona NULL pointer göndermek bir opsiyon mu?

Altın kural Word

bir türe eş isim vermenin pratik yolu

//int e eş isim olarak Word

- 1] o türden bir değişken tanımlayın
- 2] başına typedef koyun
- 3] değişken ismi yerine tür eş ismini koyun

```
typedef int Word;
```

```
typedef int* IPTR;
```

```
int main()
```

```
{  
:  
:  
}
```

```
//int[10] INTA10
```

```
typedef int INTA10[10];
```

```
int main()  
{
```

inta10

10 elemanlı dizinin eş ismi oldu

```
typedef int INTA10[10];
```

```
int main()  
{  
    INTA10 a, b, c;  
    int a[10], b[10], c;  
}
```

```
//const int * türüne CIPTR
```

```
typedef const int* CIPTR;
```

```
int main()  
{
```

Aynı türe birden fazla eş isim verilebilir.

```
typedef int Word;  
typedef int counter_t;  
typedef int Second;
```

Buda olabilir.

```
typedef int Word, counter_t, second;
```

Mülakat sorusu

Const iptr p=&x hangisidir.

```
typedef int* IPTR;

int main()
{
    int x = 10;

    const IPTR p = &x;
    //int *const p = &x;
    //const int *p = &x;
}
```

Cevap int * const p imiş herkes yanlış yaptı önemli!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

Const nesneyi betimler.

Cdec1. Org önemli site !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

Örnek multi dizi

```
typedef int INTA10[10];

int main()
{
    //INTA10 a[20];
    int a[20][10];
}
```

