

Ders 32

Tipik bir örnek nokta karakteri arama

Dosyada nokta karakteri ama sondakini bulmak için strrchr ile yapcaz

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

#define SIZE 100

int main(void)
{
    printf("bir yazi girin: ");
    char str[SIZE];
    sgets(str);

    printf("aranacak karakteri giriniz: ");
    int ch = getchar();

    char* p = strchr(str, ch);
    if (!p) {
        printf("bulunamadi\n");
    }
    else {
        printf("bulundu yazinin %d indisli karakteri (%s)\n", p - str, p);
    }

    printf("arama strrchr ile yapiliyor\n");

    p = strrchr(str, ch);
    if (!p) {
        printf("bulunamadi\n");
    }
    else {
        printf("bulundu yazinin %d indisli karakteri (%s)\n", p - str, p);
    }
}
```

Microsoft Visual Studio Debug Console

```
bir yazi girin: kahramanmaras
aranacak karakteri giriniz: a
bulundu yazinin 1 indisli karakteri (ahramanmaras)
arama strrchr ile yapiliyor
bulundu yazinin 11 indisli karakteri (as)
```

D:\KURSLAR\NISAN_2022_C\Debug\NISAN_2022_C.exe (process 32884) exited with code 0.
Press any key to close this window . . .

Örnek kendi strrchr yi yazalım

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

#define SIZE 100

char* mystrchr(const char* p, int c)
{
    char* pret = NULL;

    while (*p) {
        if (*p == c) {
            pret = (char *)p;
        }
        ++p;
    }

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

    return pret;
}
```

Bu işlemi strlen ile boyutu bulup dizinin sonundan başlayarakta arama yapabiliriz.

İdiom null karakteri gösteren pointer yazı dizisinin son-ununu

```

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

=====

while (*p)
    ++p;

=====

while (*p++)
    ; //null statement

--p;

=====

ya yazı boş ise

if (*p)
    while (*++p)
        ;

=====

p += strlen(p);

=====

p = strchr(p, '\0');

```

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

Örnek basit strcpy

```
#include "nutility.h"

#define SIZE 100

int main(void)
{
    char s1[SIZE];
    char s2[SIZE];

    printf("bir yazi girin: ");
    sgets(s1);
    ///

    strcpy(s2, s1);

    printf("(%s) (%s)\n", s1, s2);
}
```

Bu kodda s1 s2 den boyut olarak büyük olsaydı tanımsız davranış olurdu.

Burada sorumluluk kodu yazana ait yani taşma olmama garantisi ile yazmalıyız.

Örnek strcpy yi kendimiz yazalım idiomlar ile.

Kodu düzenlemeye devam edicez

```
char* mystrcpy(char* pdest, const char* psource)
{
    char* ptemp = pdest;
    while (*psource != '\0') {
        *pdest = *psource;
        ++pdest;
        ++psource;
    }

    *pdest = '\0';

    return ptemp;
}
```

```

char* mystrcpy(char* pdest, const char* psource)
{
    char* ptemp = pdest;
    while (*pdest++ = *psource++)
        ;
    return ptemp;
}

```

Çok şık bir idiom ile yazdık !!!!!!!!!!!!!!!!!!!!!

Burada atama operatörünün ürettiği değeri 0 olunca döngüden çıkacak 0 olmasında null karaktere geldiğini gösterir aynı zamanda null karakteride atayacak

Pdest değerini geri vermemiz lazım ama değiştirdik bu yüzden ptemp ile başta bu adresi aldık ve dizinin geri dönüş değeri yaptık

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

Çok önemli tanımsız davranış örneği !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

```

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

    //sevsevcan
    strcpy(str + 3, str);

    puts(str);
}

```

Microsoft Visual Studio Debug Console

sevsevcevc

Normalde sevsevcan olması lazımdı çünkü kopyalamayı 3.adresten itibaren vermiştik.

Bu bir tanımsız davranış çünkü *

Çünkü strcpy nin dökümanında belirtilen gibi çakışan adreslerde işlem yapılırsa tanımsız oluyor. Bu çakışan adresler aynı nesne veya aynı dizi elemanıdır. Bu şekilde kullanmayın.

```

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

    ///

    //str nin tuttuğu yazı volkan
    str = "volkan"
}

```

Bu şekilde yazmak sentax hatası bunu strcpy ile yapıyoruz.

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

Örnek basit strstr

```

#define SIZE 100

int main(void)
{
    char s1[100] = "bugün günlerden Salı ve C dersimiz var";
    char s2[100];

    printf("aranacak yazıyı girin: ");
    scanf("%s", s2);

    char* p = strstr(s1, s2);
    if (p) {
        printf("bulundu idx = %d (%s)\n", p - s1, p);
    }
    else {
        printf("bulunamadı\n");
    }
}

```

Salıyı aradıysak geri dönüş değeri s karakterinin adresi oluyor bunu ödev olarak kendi
Miz yazacağız.

Soru aranacak yazının yerinde harf yerine ünlem karakterleri olsun

```

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

#define SIZE 100

int main(void)
{
    char str[SIZE];
    char s[SIZE];
    printf("bir yazi girin: ");
    sgets(str); //bugun hava cok sicak

    printf("aranacak yaziyi girin: ");
    sgets(s); //hava

    //your code

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

```

Kodu yazdik

```

sgets(str); //bugun hava cok sicak

```

```

printf("aranacak yaziyi girin: ");
sgets(s); //hava

```

```

char* p = strstr(str, s);
if (p) {
    size_t len = strlen(s);
    while (len--)
        *p++ = '*';
}
else {
    printf("bulunamadi\n");
}

```

```

printf("(%s)\n", str); // (bugun **** cok guzel)

```

//

Örnek basit strcat

```
int main(void)
{
    char s1[SIZE];
    char s2[SIZE];

    printf("iki yazi giriniz: ");
    scanf("%s%s", s1, s2); //cemal  dagli

    printf("(%s) (%s)\n", s1, s2); // (cemal) (dagli)
    strcat(s1, s2);
    printf("(%s) (%s)\n", s1, s2); // (cemaldagli) (dagli)
}
```

Bir yazının sonuna diğer yazıyı ekliyor.

Değişen yazının adresi döner

//

Örnek kendimiz strcat yazalım

```
char* mystrcat(char* p1, const char* p2)
{
    char* ptemp = p1;

    while (*p1)
        ++p1;

    while (*p1++ = *p2++)
        ; //null statement

    return ptemp;
}
```

Bugünkü 2 idiomu kullandık.

```

char* mystrcat(char* p1, const char* p2)
{
    strcpy(p1 + strlen(p1), p2);
    return p1;
}

```

Buda std fonksiyonları kullanarak kesin tekrar et çok önemli

```

char* mystrcat(char* p1, const char* p2)
{
    strcpy(strchr(p1, '\0'), p2);
    return p1;
}

```

Strchr bize null karakterinin adresini vericek bizimde amacımız bu zaten

Yazıyı nullo'dan sonra eklemiş olduk.

Taşma olmaması gerekir bunu programcı dikkat edicek.

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

Örnek 2 yazıyı başka bir dizide birleştir.

```

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

#define SIZE 1000

int main(void)
{
    char s1[SIZE];
    char s2[SIZE];
    char s3[SIZE];

    printf("iki yazi giriniz: ");
    scanf("%s%s", s1, s2); //cemal dagli

    printf("(%s) + (%s) = (%s)\n", s1, s2, s3); // (cemal) + (dagli) = (cemaldagli)
}

```



```
#define SIZE 1000
```

```
int main(void)
{
    char s1[SIZE];
    char s2[SIZE];
    char s3[SIZE];

    printf("iki yazı giriniz: ");
    scanf("%s%s", s1, s2); //cemal   dagli

    strcpy(s3, s1);
    strcat(s3, s2);

    printf("(%s) + (%s) = (%s)\n", s1, s2, s3); // (cemal) + (dagli) = (cemaldagli)
}
```

Sadece 2 kod yazdık strcpy ile s1 i s3 e kopyaladık.

Strcat ilede s3 sonuna s2 yi yazdırdık.

```
char s1[SIZE];
char s2[SIZE];
char s3[SIZE];

printf("iki yazı giriniz: ");
scanf("%s%s", s1, s2); //cemal   dagli

strcat(strcpy(s3, s1), s2);

printf("(%s) + (%s) = (%s)\n", s1, s2, s3); // (cemal) + (dagli) = (cemaldagli)
```

İlk çağrıda str cat olsaydı çalışırmıydı. Bence çalışmaz.

Run time hatası çünkü s3 de çöp değer var null karakter yok otomatik ömürlü bunlar.

```

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

#define SIZE 1000

int main(void)
{
    char s1[SIZE];
    char s2[SIZE];
    char s3[SIZE];

    printf("iki yazi giriniz: ");
    scanf("%s%s", s1, s2); //cemal dagli

    s3[0] = '\0';
    strcat(s3, s1);
    strcat(s3, s2);

    printf("(%s) + (%s) = (%s)\n", s1, s2, s3); // (cemal) + (dagli) = (cemaldagli)
}

```

İlk değer verirse olur ='' yada null ile bide global yapmayı deneyiyim.

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

Yazıların karşılaştırılması.

fonksiyon karşılaştırma işleminin kesin sonucunu iletiyor

eray ertan

int

birinci büyük ise fonksiyon pozitif bir deger döndürecek

ikinci büyük ise fonksiyon negatif bir deger döndürecek

ikisi eşit ise fonksiyon 0 döndürecek

cumhuriyet ok

lexicographical compare
=====

oge sayilari esit
karsilikli tum ogeler esit

karşılıklı elemanlar çift çift karşılaştırıldığında
ilk farklı çiftten büyük olanına sahip olan büyüktür

```
int a[] = {5, 9};  
int b[] = {1, 12},,,}
```

Burda a büyük ilk farklı olan elemanın büyüğü büyüktür.

```
char s1[SIZE];  
char s2[SIZE];  
  
if (s1 != s2)  
  
if (strcmp(s1, s2) != 0)  
if (strcmp(s1, s2))  
  
if (strcmp(s1, s2) == 0)  
if (!strcmp(s1, s2))  
  
if (strcmp(s1, s2) > 0)  
if (strcmp(s1, s2) < 0)
```

En çok 4. Yazılan karışık ! değil ile eşit olduğuna bakıyoruz çünkü eşit ile 0 değeri geri dönüyor.

!0 da 1 olduğuna göre eşitse if(!strcmp(s1,s2))

Basit örnek

```

int main(void)
{
    char s1[SIZE];
    char s2[SIZE];

    printf("iki kelime girin: ");
    scanf("%s%s", s1, s2);

    int result = strcmp(s1, s2);

    if (result > 0)
        printf("(s1) > (s2)\n", s1, s2);
    else if (result < 0)
        printf("(s1) < (s2)\n", s1, s2);
    else
        printf("(s1) = (s2)\n", s1, s2);
}

```

Microsoft Visual Studio Debug Console

```

iki kelime girin: can cancel
(can) < (cancel)

```

Microsoft Visual Studio Debug Console

```

iki kelime girin: ankara antalya
(ankara) < (antalya)

```

Microsoft Visual Studio Debug Console

```

iki kelime girin: masa MASA
(masa) > (MASA)

```

Eşitlik sınaması

```

int main(void)
{
    char s1[SIZE];
    char s2[SIZE];

    printf("iki kelime girin: ");
    scanf("%s%s", s1, s2);

    if (!strcmp(s1, s2))
        printf("evet esit\n");
}

```

Kendi strcmp yazalım

```
int mystrcmp(const char* p1, const char* p2)
{
    while (*p1 == *p2) {
        if (*p1 == '\0')
            return 0;
        ++p1;
        ++p2;
    }

    return *p1 - *p2;
}
```

Eşitlikten farklı karakterler olunca çıkacak

P1 büyükse pozitif

P2 büyükse negatif

Eşitlik içinde null sorgulamasına girerse 0 dönecek.