

## Örnek strncpy tanımsız örnek

```

#define      SIZE      100

int main(void)
{
    char s1[SIZE];
    char s2[SIZE] = "ERAY GOKSU";

    size_t n;

    printf("kac karakter kopyalanacak: ");
    scanf("%zu", &n);

    strncpy(s1, s2, n);

    puts(s1);
}

```

Bu örnekte null karakter yazmadığı için sıkıntı olabilir. N dizi yazı boyutundan küçükçe ekrana yazdırırken puts tanımsız davranış olur.

```

strncpy(s1, s2, n);
s1[n] = '\0';

puts(s1);

```

Bu şekilde sonuna null koyduk

Aslında bu memcpy ye benziyor.

```

scanf("%zu", &n);

strncpy(s1, s2, n)[n] = '\0';

puts(s1);

```

Çok güzel bir idiom geri dönüş değeri yazının yazıldığı dizinin ilk adresi bunu n inci indexi ile çağırıp ona null atadım.

P[n] = '\n' dizinin n inci elemanına null atadım çok güzel yazım.

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

Örnek neden null kopyalanmıyor

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

#define SIZE 100

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

    strncpy(str + 2, "KUT", 3);
    puts(str);
}
```

Çünkü null koysaydık ekrana Erkut yazardı ancak biz yazının belli bir kısmını değiştirmek istiyoruz.

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

Strncat

Örnek

```
int main(void)
{
    char s1[SIZE] = "murat";
    char s2[SIZE] = "aksakal";

    strncat(s1, s2, 3)

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

Ekranda murataks yazacak tanımsız davranış yok yazının sonuna nula koyuyor.

N 5 olsaydı murataksak yazardı.

N8 olsaydı murataksakal yazardı hatta printf içindede yazılırdı.

```
#define      SIZE      100      I
int main(void)
{
    char s1[SIZE] = "murat";
    char s2[SIZE] = "aksakal";

    printf("(s)\n", strncat(s1, s2, 5));
}
```

Fonksiyonun geri dönüş değeri

Yazının yazıldığı dizinin adresi olduğu için printf içindede yazılabilir.

1.10 dan öncesini not al

1.33 den sonrası not alındı.

Sprintf bildirimi

```
int printf(const char *pfm, ...);
int sprintf(char *pstr, const char *pfm, ...);
int fprintf(FILE *f, const char *pfm, ...);
```

Örnek sprintf

Ekranda normalde yazılabileceğimiz bir yazıyı başka bir diziye yazdırabilmek.

```

int main(void)
{
    int x;

    printf(_Format: "bir sayi girin: ");
    scanf(_Format: "%d", &x);
    char name[SIZE];
    printf(_Format: "ismi girin: ");
    scanf(_Format: "%s", name);

    char file_name[SIZE]; //ertan_8712.jpg
    printf(_Format: "%s_%d.jpg", name, x);
}

```

```

#define _CRT_SECURE_NO_WARNINGS
Microsoft Visual Studio Debug Console
bir sayi girin: 4567
ismi girin: harun
harun_4567.jpg
D:\KURSLAR\NISAN_2022_C\Debug\NISAN_2022_C.exe (process 20712) exited with code 0.
Press any key to close this window . . .

```

Burada yazdırmak istediğimizi kolay bir şekilde ekrana yazdırdık şimdi sprintf kullanıcaz printf yerine

```

int main(void)
{
    int x;

    printf("bir sayi girin: ");
    scanf("%d", &x);
    char name[SIZE];
    printf("ismi girin: ");
    scanf("%s", name);

    char file_name[SIZE]; //ertan_8712.jpg
    sprintf(file_name, "%s_%d.jpg", name, x);

    printf("dosya ismi: %s\n", file_name);
}

```

```

Microsoft Visual Studio Debug Console
bir sayi girin: 7564
ismi girin: eray
dosya ismi: eray_7564.jpg
D:\KURSLAR\NISAN_2022_C\Debug\NISAN_2022_C.exe (process 10288) exited with
Press any key to close this window . . .

```

Bu kadar kolay

Örnek gerçek sayıları yazıya çevirme

E bunun için en kolay yol sprintf ile yazıyı diziye yazdırmak

```
int main(void)
{
    double dval;
    printf("bir gercek sayi giriniz: ");
    scanf("%lf", &dval);

    char str[SIZE];

    sprintf(str, "%.3f", dval);

    printf("(%.3s)\n", str);
}
```



Microsoft Visual Studio Debug Console

bir gercek sayi giriniz: 98435.87345  
(98435.873)

Printf ile ekrana yazdırmak gibi tek farkı yazılacak adresi ilk argüman ile char dizi adresi vermem çok güzel bir fonksiyon

Örnek sprintf dosya isimleri oluşturma

```

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

#define SIZE 100

int main(void)
{
    ///
    char file_name[SIZE];
    for (int i = 1; i < 123; ++i) {
        sprintf(_Buffer:file_name, _Format:"ertan%03d.txt", i);
        printf(_Format:"%s\n", file_name);
    }
}

```

```

#define _CRT_SECURE_NO_WARNINGS

```

Microsoft Visual Studio Debug Console

```

ertan089.txt
ertan090.txt
ertan091.txt
ertan092.txt
ertan093.txt
ertan094.txt
ertan095.txt
ertan096.txt
ertan097.txt
ertan098.txt

```

Her i için farklı bir isim oluşturdum ve bunu file\_names dizisine yazdırdım.

```

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

```

Scanf

Örnek sscanf

```

int main(void)
{
    int ival;
    printf("bir tamsayi girin: ");
    int result = scanf("%d", &ival);

    printf("ival = %d\n", ival);
    printf("result = %d\n", result);
}

```

```

C:\Microsoft Visual Studio Debug Console
C:\bir tamsayi girin: 873245
C\ival = 873245
C\result = 1

```

Geri dönüş değeri başarılı olan set miktarı. Bu basit scanf ile Sscanf ile yazalm.

```

int main(void)
{
    char buffer[] = "8764235ali";
    int ival;

    sscanf(buffer, "%d", &ival);

    printf("ival = %d\n", ival);
}

```

```

Microsoft Visual Studio Debug Console
ival = 8764235
D:\KURSLAR\NISAN_2022_C\Debug\NISAN_2022_C.exe (process 26116)
Press any key to close this window . . .

```

Ekranda nasıl sadece sayıyı alıp yazıyı bırakıyorsa burdada aynı diziden ilk gördüğü sayıyı alıyor scanf ile aynı kurallar.

Boşluklu ve 4 değişken olsaydı.

```

#define SIZE 100

int main(void)
{
    char buffer[] = "912 376 78345 456";
    int x, y, z, t;

    sscanf(buffer, "%d%d%d%d", &x, &y, &z, &t);

    printf("x = %d\n", x);
    printf("y = %d\n", y);
    printf("z = %d\n", z);
    printf("t = %d\n", t);
}

```

```

Microsoft Visual Studio Debug Console

x = 912
y = 376
z = 78345
t = 456

D:\KURSLAR\NISAN_2022_C\Debug\NISAN_2022_C.exe (process 23052) exited with code

```

Dizinin içindeki boşluk separatör görevi görür

Çok güzel bir örnek

912ahmet yazısının 912 yi bir değişkene atarız ise bir diziye aktarılması.

```

int main(void)
{
    char buffer[] = "912ahmet";

    char name[20];
    int ival;
    sscanf(buffer, "%d%s", &ival, name);

    printf("ival = %d\n", ival);
    puts(name);
}

```

```

Microsoft Visual Studio Debug Console

ival = 912
ahmet

D:\KURSLAR\NISAN_2022_C\Debug
Press any key to close this w

```

912 de ival duracak kalan kısmı yazı olduğu için name dizisi alıcak scanf ile aynı ya sadece yazıyı aldığı yeri kendimiz dizide veriyoruz.

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



```
<stdlib.h>
```

```
    exit
```

```
atexit
```

```
    abort
```

I

Örnek exit.

```
void f4(void)
{
    printf("f4 cagrildi\n");
    f4();
    printf("f4 sona erdi\n");
}

void f3(void)
{
    printf("f3 cagrildi\n");
    f4();
    printf("f3 sona erdi\n");
}

void f2(void)
{
    printf("f2 cagrildi\n");
    f3();
    printf("f2 sona erdi\n");
}
```

```

void f1(void)
{
    printf("f1 cagrildi\n");
    f2();
    printf("f1 sona erdi\n");
}

int main(void)
{
    printf("main cagrildi\n");
    f1();
    printf("main sona erdi\n");
}

```

```

Microsoft Visual Studio Debug Console
main cagrildi
f1 cagrildi
f2 cagrildi
f3 cagrildi
f4 cagrildi
f4 sona erdi
f3 sona erdi
f2 sona erdi
f1 sona erdi
main sona erdi

D:\KURSLAR\NISAN_2022_C\Debug\NISAN_2022_C.exe (pro
Press any key to close this window . . .

```

Çaşırtınca ekran çıktısı bu olur

Hepsi iç içe çağırılacak sonra sıra ile sonlanacak.

Ama programı f4 de bitirmek isteseydik.

```

void f4(void)
{
    printf("f4 cagrildi\n");
    exit(EXIT_FAILURE);
    printf("f4 sona erdi\n");
}

void f3(void)
{
    printf("f3 cagrildi\n");
    f4();
    printf("f3 sona erdi\n");
}

```

```

Microsoft Visual Studio Debug Console
main cagrildi
f1 cagrildi
f2 cagrildi
f3 cagrildi
f4 cagrildi

D:\KURSLAR\NISAN_2022_C\Debug\NISAN_2022_C.exe (process 10036) exited with code 1.
Press any key to close this window . . .

```

Normal olarak proses sonlandı ama mainde değil f4 içinde

Argüman olarak failer ile succes geçmek arasında program sonlandırma arasında burda fark yok

Ama 0 ise başarılı ama 1 olsaydı başarısız olacaktı.

```
int main(void)
{
    printf("main cagrildi\n");
    f1();
    printf("main sona erdi\n");
    exit(-87325);
}
```

Mainde return ile exit ile bitirmek aynı.

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

```
}

void foo()
{
    exit(EXIT_FAILURE);
}

int main(void)
{
    ///....
    atexit(f1);
    atexit(f2);
    atexit(f3);
    atexit(f4);

    foo();
}
```

```
Microsoft Visual Studio Debug Console

f4 cagrildi
f3 cagrildi
f2 cagrildi
f1 cagrildi

D:\KURSLAR\NISAN_2022_C\Debug\NISAN_2022_C.exe (process 28368) exited with code 1.
Press any key to close this window . . .
```

Kayıt ettiğim fonksiyonlar sıra ile çağırılacak

Ve aynı fonksiyonu birden fazla kayıt edebiliriz.

Soru foo yu sildim main i boş yazdım.

Yinede f1 f2 ..... çağırılacak mı

```
printf(_Format: "f1 cagrildi\n");  
}  
  
int main(void)  
{  
    ///....  
    atexit(f1);  
    atexit(f2);  
    atexit(f3);  
    atexit(f4);  
}
```

Evet çünkü main kendi içinde exit fonksiyonunu ile programı bitiriyor.