ITU ACM Student Chapter Course Program

Introduction to C

Week 6

Instructor

Mehmet Yiğit Balık

Prepared by

Mehmet Yiğit Balık & Mihriban Nur Koçak & Emir Oğuz

Pointers and Strings (Göstericiler ve Karakter Dizileri)

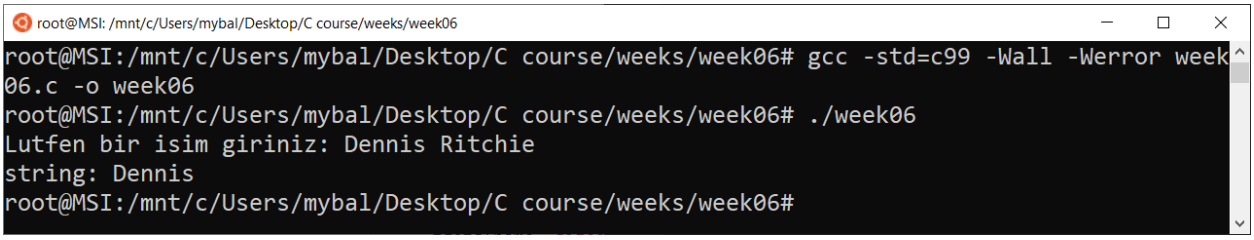
Önceki derslerde string'lerin birer karakter dizisi oldukları ve dizilerin aslında pointerlar şeklinde ifade edilebildiklerinden bahsedilmişti. Bu bilgilerden yola çıkarak string'lerin pointerlar yoluyla kolaylıkla ifade edilebilen yapılar olduğu sonucuna varılabilir.

```c
#include <stdio.h>
#include <stdlib.h>

int main()
{
    //Asagidaki iki farkli string atamasi esasinda ayni islevi gorur
    char string1[] = "Dennis Ritchie";
    char *string2 = "Dennis Ritchie";
    printf("string1: %s\n", string1);
    printf("string2: %s\n", string2);
    return EXIT_SUCCESS;
}
```

```
root@MSI: /mnt/c/Users/mybal/Desktop/C course/weeks/week06
root@MSI:/mnt/c/Users/mybal/Desktop/C course/weeks/week06# gcc -std=c99 -Wall -Werror week
06.c -o week06
root@MSI:/mnt/c/Users/mybal/Desktop/C course/weeks/week06# ./week06
string1: Dennis Ritchie
string2: Dennis Ritchie
root@MSI:/mnt/c/Users/mybal/Desktop/C course/weeks/week06#
```

Kullanıcıdan girdi olarak bir string almak için formatlama olarak **%s** kullanılır fakat **scanf** içerisinde formatlamadan sonra string'e bir değer atamak için **&** işareti kullanılmaz. **&** işaretinin değişkenin adresini belirttiği önceki derslerde gösterilmişti. Bu durumda string'ler esasında pointer'lar işlevinde kullanılabilen yapılar oldukları için bir değer atanırken **&** kullanılmasına gerek yoktur.

```c
#include <stdio.h>
#include <stdlib.h>

int main()
{
    //Asagidaki iki farkli string atamasi esasinda ayni islevi gorur
    char string[100];
    printf("Lutfen bir isim giriniz: ");
    scanf("%s", string);
    printf("string: %s\n", string);
    return EXIT_SUCCESS;
}
```



Yukarıda görüldüğü üzere girdi olarak *Dennis Ritchie* verilmesine rağmen sadece *Dennis* bastırılmıştır. Bunun sebebi **scanf** fonksiyonunun girdiyi sadece boşluk gördüğü noktaya kadar almasıdır. Bunun önüne geçmek için bazı fonksiyonlara ve string kütüphanesine değinilmelidir.

fgets(), puts()

fgets() fonksiyonu ilerideki derslerde de gösterilecek olan bir girdi alma fonksiyonudur. Bu derste sadece aşina olunan terminal yoluyla girdi alınacaktır.

Yukarıdaki örnekte olduğu gibi *Dennis Ritchie* girdisi **scanf** ile alınmak istendiğinde, **scanf** bu kelime grubunu boşluğa kadar okur. **fgets** ise **\n**'e kadar yani kullanıcının girdiği satırın sonuna kadar olan kısmı girdi olarak alır.

fgets(char * var,sizeof(char * var),**stdin)**

Bu fonksiyonun aldığı ilk parametre kullanıcının girdiği değerin atanacağı değişkendir yani bir karakter dizisidir, ikinci parametre ise kullanıcının girdiği değerin maksimum büyüklüğüdür. Üçüncü değişken olarak, eğer girdi INPUT'dan alınmak isteniyorsa **stdin (standard input)** yazılmalıdır. Üçüncü parametre bahsedildiği gibi ilerleyen konularda farklı formatlarda gösterilecektir.

puts() fonksiyonu neredeyse **printf()** ile aynı işlevi görür fakat **printf()**'den farklı olarak **puts()** fonksiyonu sadece **string**'ler için çalışır ve yazdırdığı **string**'in sonuna otomatik olarak **\n** koyar.

puts(char *)

```c
#include <stdio.h>
#include <stdlib.h>

int main()
{
    char string[100];
```

```c
    printf("Lutfen bir isim giriniz: ");
    fgets(string, sizeof(string), stdin);
    puts(string);
    return EXIT_SUCCESS;
}
```

```
root@MSI: /mnt/c/Users/mybal/Desktop/C course/weeks/week06
root@MSI:/mnt/c/Users/mybal/Desktop/C course/weeks/week06# gcc -std=c99 -Wall -Werror week
06.c -o week06
root@MSI:/mnt/c/Users/mybal/Desktop/C course/weeks/week06# ./week06
Lutfen bir isim giriniz: Dennis Ritchie
Dennis Ritchie

root@MSI:/mnt/c/Users/mybal/Desktop/C course/weeks/week06#
```

String Kütüphanesi <string.h>

String kütüphanesi diğer kütüphaneler gibi kaynak dosyasının üst tarafında diğer kütüphaneler ile birlikte yazılmalıdır. Bu kütüphanenin içinde barındırdığı başlıca fonksiyonlar:

- **strlen(**char* variable**)**: Bir string'in uzunluğunu hesaplar.

- **strcpy(**char* destination, const char* source**)**: Bir string'i başka bir string'e kopyalar.

  ***Not**: String`ler birbirine direkt = yoluyla atanamaz. Bu noktada **strcpy()** fonksiyonunun kullanımı çok büyük önem taşır.*

- **strcat(**char* destination, const char* source**)**: Bir string'i başka bir string'in sonuna ekler.

- **strcmp(**const char* str1, const char* str2**)**: İki string'i birbirleri ile karşılaştırır.
  - Eğer 0 döndürürse bu iki string aynıdır.
  - Eğer negatif bir sayı döndürürse **str1** alfabetik olarak **str2**'den daha önce geliyordur.

- Eğer pozitif bir sayı döndürürse **str2** alfabetik olarak **str1**'den daha önce geliyordur.

```c
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
int main()
{
    char string[100] = "Dennis Ritchie";
    int uzunluk = strlen(string);
    printf("%s'in uzunlugu: %d\n", string, uzunluk);
    char string2[100];
    strcpy(string2, string);
    puts(string2);
    char string3[100] = "My name is ";
    strcat(string3, string);
    puts(string3);

    char compare1[100] = "abcd";
    char compare2[100] = "abcd";
    char compare3[100] = "wxyz";
    // 0 donme durumu
    printf("%d\n", strcmp(compare1, compare2));
    // negatif donme durumu
    printf("%d\n", strcmp(compare1, compare3));
    // pozitif donme durumu
    printf("%d\n", strcmp(compare3, compare1));
    return EXIT_SUCCESS;
}
```

```
Select root@MSI: /mnt/c/Users/mybal/Desktop/C course/weeks/week06
root@MSI:/mnt/c/Users/mybal/Desktop/C course/weeks/week06# gcc -std=c99 -Wall -Werror week
06.c -o week06
root@MSI:/mnt/c/Users/mybal/Desktop/C course/weeks/week06# ./week06
Dennis Ritchie'in uzunlugu: 14
Dennis Ritchie
My name is Dennis Ritchie
0
-22
22
root@MSI:/mnt/c/Users/mybal/Desktop/C course/weeks/week06#
```

Pointers and Functions (Göstericiler ve Fonksiyonlar)

Şu ana kadar derslerde gösterilen tüm fonksiyon örneklerinde fonksiyonlara parametre olarak gönderilen değişkenlerde, fonksiyondan çıktıktan sonra bir değişiklik olmadığı gözlemlenebilir. Çünkü fonksiyonlar parametre olarak aldıkları değerleri önce kendi içlerinde oluşturdukları yerel değişkenlerine atayarak işleme sokarlar. Bu sebeple fonksiyon içerisinde aslında işlem gören değişkenler, fonksiyona parametre olarak gönderilen değişkenler değil fonksiyonun kendi yerel değişkenleridir. (**pass by value**)

Burada akla gelebilecek ilk soru bir değişkenin bir fonksiyon sonucu direkt olarak değerinin nasıl değiştirilebileceğidir. İşte bu noktada pointer'lar devreye girer. Eğer bir fonksiyon parametre olarak bir değişkenin değerini değil de değişkenin adresini alırsa, bu fonksiyonda bir yerel değişken oluşturulmaz, direkt adresi gönderilen değişken fonksiyon içerisinde işlem görür. Fonksiyon içerisinde işlem gören kısım bu adresin bulunduğu kısım olacaktır ve bu şekilde bir değişkenin değeri fonksiyon içerisinde değiştirilebilecektir. (**pass by reference**)

```c
#include <stdio.h>
#include <stdlib.h>

void swap_by_value(int a, int b)
{
    int temp = a;
    a = b;
    b = temp;
}
void swap_by_reference(int *a, int *b)
{
    int temp = *a;
    *a = *b;
    *b = temp;
}

int main()
{
    int x = 15;
    int y = 3;
    swap_by_value(x, y);
    printf("x'in degeri degismedi: %d\ny'nin degeri degismedi: %d\n", x, y
);

    swap_by_reference(&x, &y);
    printf("x'in yeni degeri: %d\ny'nin yeni degeri:%d\n", x, y);

    return EXIT_SUCCESS;
}
```

```
root@MSI: /mnt/c/Users/mybal/Desktop/C course/weeks/week06
root@MSI:/mnt/c/Users/mybal/Desktop/C course/weeks/week06# gcc -std=c99 -Wall -Werror week
06.c -o week06
root@MSI:/mnt/c/Users/mybal/Desktop/C course/weeks/week06# ./week06
x'in degeri degismedi: 15
y'nin degeri degismedi: 3
x'in yeni degeri: 3
y'nin yeni degeri:15
root@MSI:/mnt/c/Users/mybal/Desktop/C course/weeks/week06#
```

String'ler pointer'larla ifade edilebilen yapılar oldukları için fonksiyonlara parametre olarak gönderildiklerinde aslında string'in adresi olarak gönderilmiş olurlar.

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

void hello(char *name)
{
    char greeting[100] = "Hello ";
    strcat(greeting, name);
    puts(greeting);
}

int main()
{
    char name[100];
    printf("Lutfen isminizi giriniz: ");
    fgets(name, sizeof(name), stdin);
    hello(name);
    return EXIT_SUCCESS;
}
```

```
root@MSI: /mnt/c/Users/mybal/Desktop/C course/weeks/week06
root@MSI:/mnt/c/Users/mybal/Desktop/C course/weeks/week06# gcc -std=c99 -Wall -Werror week
06.c -o week06
root@MSI:/mnt/c/Users/mybal/Desktop/C course/weeks/week06# ./week06
Lutfen isminizi giriniz: Mark Zuckerberg
Hello Mark Zuckerberg

root@MSI:/mnt/c/Users/mybal/Desktop/C course/weeks/week06#
```