**BLGM 112 – DENEY 6[[1]](#footnote-2)\***

# İşaretçiler

## Amaçlar

1. İşaretçilerin temel özellikleri ve kullanımlarının tanıtımı

## Ön bilgi

İşaretçiler, bellek üzerindeki herhangi bir konuma erişebilen özel değişkenlerdir. İşaretçiler, temelde diğer değişkenlerden farklı değillerdir, fakat barındırdıkları değer bir hafıza adresidir. C Programlama Dili'nde, işaretçilerin verimli kullanımı için bir takım araçlar bulunmaktadır. Bunlardan ilki adres operatörüdür (&). Adres operatörü tek bir terim (operand) alır ve bu terimin sol tarafında bulunur. Adres operatörünün en çok görüldüğü yerlerden biri scanf fonksiyonudur. Diğer bir operatör ise herhangi bir işaretçinin değerini veren \* operatörüdür. Bu operatör de, tıpkı adres operatörü gibi, herhangi bir işaretçinin sol tarafında kullanılarak, o işaretçinin gösterdiği adresteki değeri verir. İşaretçiler, diğer değişkenlerde farklı tanımlanırlar. Aşağıda bir işaretçi tanımı verişmiştir.

int \*p;

İşaretçiler tipiden bağımsız değillerdir. Yani, *int*\* tipinde bir işaretçi yalnızca *int* tipinde bir değişkenin adresini gösterebilir veya *float* tipinde bir adrese erişmek isteniliyorsa, işaretçi *float\** şeklinde tanımlanmalıdır. Aşağıda, bahsedilen operatörlerin kullanımları gösterilmiştir.

#include <stdio.h>

int main() {

int sayi = 2;

// p, sayi değişkeninin adresini işaret eder

int \*p = &sayi;

// p işaretçisinin değerini ekrana yazdır

printf(“sayı = %d\n”, \*p);

return 0;

}

Tıpkı sıradan değişkenler gibi, işaretçiler üzerinde de birtakım aritmetik işlemler tanımlıdır. Bunlar,

* +
* -
* ++
* --

İşaretçiler, sıradan değişkenlerin aksine, hafıza ile doğrudan ilişkili olduğu için bu işlemlerin çok dikkatli kullanılması gerekmektedir. Örneğin, bir işaretçinin değerinin bir arttırılması, bu işaretçinin bir sonraki hafıza adresini göstermesini sağlar. Bu durumda artış miktari işaretçinin tipine göre değişmektedir. Bir değişken boyutu 4 byte ise (örneğin, i*nt)*, bir sonraki aynı tiptde değişkenin bulunabileceği tek yer 4 byte sonrakı adrestir. Yani, *int* tipinde bir işaretçiye 1 eklenirse, değeri 4 artacaktır. Benzer olarak, *char* tipindeki değişkenler hafızada 1 byte olarak tutulduğu için, bu tipdeki işaretçinin bir arttırılması, değerinin 1 artmasını sağlar.

Bir işaretçi, başka bir işaretçinin bulunduğu bir adresi de gösterebilir. Bunun için işaretçinin uygun tipde tanımlanması yeterlidir. Aşağıda, bu duruma örnek tanımlamalar verilmiştir.

int sayi = 2;

int \*p1 = &sayi; //int tipinde işaretçi

int \*\*p2 = &p1; //işaretçiyi işaret eder

int \*\*\*p3 = &p2; //işaretçiyi işaret eden işaretçiye işaret eder

// ...

printf(“sayı = %d,p1 = %d,p2 = %d,p3 = %d”, sayi, \*p1, \*\*p2, \*\*\*p3);

## Deneyler

1. Verilen iki sayının değerlerini birbiriyle değiştiren iki fonksiyon yazınız. Bu fonksiyonlardan bir tanesi, işaretçi; diğeri ise sıradan değişkenler kabul etsin. İki fonksiyonu çalıştırıp, işaretçi kullanmanın nasıl bir etkisi olduğunu gözlemleyiniz. Aşağıdaki fonksiyon prototiplerini kullanabilirsiniz.

void degis1(int sayi1, int sayi2);

void degis2(int \*sayi1, int \*sayi2);

1. Farklı tipde işaretçiler tanımlayınız ve üzerlerinde yukarıda belirtilen aritmetik işlemleri uygulayarak sonuçları gözlemleyiniz.

1. \* BLGM 112 dersi için Bahar 2014/15 döneminde Cengiz Kandemir ve Cem Kalyoncu tarafından hazırlanmıştır. [↑](#footnote-ref-2)