

İST 205 Bilgisayar Programlama III
C Programlamaya Giriş ve Matematiksel-İstatistiksel Uygulamalar
Y.Doç.Dr. Levent Özbek
Ankara Üniversitesi Fen Fakültesi İstatistik Bölümü
Tel: 0.312.3126720/1420
ozbek@science.ankara.edu.tr
Web: <http://80.251.40.59/science.ankara.edu.tr/ozbek/>

1. Bölüm

DERLEYİCİLER VE C DİLİNİN ÖZELLİKLERİ

Üst düzey dillerden biri ile yazılmış olan bir programı, makine diline çeviren programa derleyici denir. C++ da böyle bir derleyicidir.

C dilini diğerlerinden ayıran özellikleri şöyle sıralayabiliriz.

- a) C dili orta sınıf bir dildir.
- b) C dili yapısal bir dildir.
- c) C dili gerçek programcıların kabul ettiği bir dildir.
- d) C dili değişik bilgisayarlara kolayca adapte olan bir dildir.

C, genel amaçlı bir dil olarak hazırlanmıştır. PC'nin, BASIC veya PASCAL gibi diller kullanılarak ulaşılamayan bazı bölümlerin kullanılmasını sağladığı için, uygulama programcılarının birçoğu C'yi yeğlerler. C, bilgisayarın herhangi bir yerine kolayca ulaşmamızı sağlar.

C kullanırken yapabileceğimiz en önemli hatayı, yalnızca bilgisayarı kapatıp açarak düzeltebiliriz. C, PC'nin ortaya çıkması ile başlamadı. C'nin kökeni çok daha eskiye gider. C, 70'li yılların başında, AT&T tarafından, mini bilgisayarlarda kullanılmak üzere tasarlanmıştır. Bu dilin beğenilmesi sonucunda, C kısa zamanda yayıldı. Sonunda C, UNIX adı verilen bir işletim sisteminin oluşturulmasında kullanıldı. 1978'de American National Standards Institute (ANSI) tarafından standart hale getirilen C dili orta sınıf bir dildir.

C'YE GENEL BİR BAKIŞ

Yapısal programlama, program karmaşıklığını azaltmak ve programları daha kolay anlaşılabilir, okunabilir, geliştirilebilir, değiştirilebilir biçime getirmeyi amaçlayan genel bir programlama yöntemidir.

Metin kütüğünü yazmak için, bir metin editörü veya bir kelime işlemci kullanılmalıdır. Derleyicilerin pek azının kendi editörleri vardır.

C ÇEVRESİ

C çevresi, editör ve derleyiciyi tek bir programda birleştirdiği için, programın yazılmasını ve edit edilmesini kolaylaştırır.

Bir programı oluşturmak için, C editörüne girin ve program metnini yazın. Programın tümünü yazdıktan sonra, C derleyicisini doğrudan editörden çalıştırabilirsiniz. Programı derlenmeden önce diske/diskete kaydetmeniz gerekli değildir. Tek bir tuşa basarak programı derleyebilirsiniz.

Üzerinde çalıştığınız kütük, bellekte tutulduğu için, birleşik çevre derleme işini bütün derleyicilerden daha hızlı yapar. Ayrı bir editör ve derleyici yerine tek bir program kullanmanın hızlı olmanın yanında başka avantajları da vardır.

Birleşik C çevresinde, editör derleyicinin bulduğu hataları her zaman bilir. C editörü, derleyicinin bulduğu her hatanın yerini size gösterir.

Birleşik C çevresi kullanıldığında oluşan yapı;

- programın yazılması
- programın işletilmesi
- programın değiştirilmesi
- programın işletilmesi

MENÜ SİSTEMİ:

C değişik bir menü sistemi kullanır. Menüler programlama sırasında sık sık kullanılırlar. Bir menü her zaman, ne yapabileceğinizi çok açık biçimde gösterir.

C PROGRAM YAPISI

TÜMCELER : Bütün C programları, bir dizi tümceden oluşurlar. Genellikle bir tümce tek bir eylem olarak düşünülebilir. Tümceler bir dizi sözcükten (komuttan) oluşurlar. Her tümce C'de ";" karakteri ile bitirilir.

ÖRNEK

```
int i;  
a=a+1;  
while(z<23) a=a+54;
```

Her satıra tek bir tümce yazılırsa, tümceleri birbirinden ayırmak daha kolay olur. Bu derleyici için bir zorunluluk değildir.

Tümceler genellikle iki gruptan birine girerler.

- işletilebilen
- bildirim

İşletilebilen tümceler, işlem yapabilen komutlar içerirler. Bu tür tümcelerin bir blok içine ({ } parantezleri içine) yerleştirilmeleri gerekir.

Bildirim tümceleri, programın işletmek zorunda olduğu yapıları belirlerler. Aşağıdaki özel bildirim tümcesi, bütün C programlarında bulunmalıdır.

Main()

Bu tümce, programın nerede çalışmaya başlayacağını gösterir.

BLOKLAR

Tümceler, blok adı verilen birimlerde bir araya toplanırlar. Bir blok, herhangi bir tümce veya blok grubundan oluşabilir. Blokun başlangıcını ve bitişini göstermek için oklu parantezler ({ }) kullanılır. Blokları birbirinden ayırmak, programı anlamak açısından oldukça önemlidir.

ÖRNEK

```
z=z*x /* blok 1 */
{ /* blok 2 */
printf("%d", z);
for (i=5;i<z;i++);
printf("%d",z);
}
{ /* blok 3 */
i=i-1;
d=z*x-i;
printf("%d", d);
}
{ /* blok 4 */
{ /* blok 4-a */
{ /* blok 4-b */
i=78;
i=i*z;
printf("%d", i);
}
printf("%d", z);
}
}
```

DERLEME KOMUTLARI

Bir C programı, tümcelere ek olarak, derleyici için komutlar da içerebilir. Derleyici komutları adı verilen bu komutlar, gerçek programın bir parçası değildir. Programda bulunan diğer tümcelerin çevirisini yönetirler.

Bu komutlar her zaman numara sembolü (#) ile başlarlar. Programın herhangi bir yerinde, tek bir satırda, yalnız başlarına olmalıdırlar.

AÇIKLAMALAR

Açıklamalar herhangi bir yerde olabilir, /* karakterleri ile başlamalı ve */ karakterleri ile bitmelidir.

ÖRNEK

```
# include <stdio.h> /* standart girdi/çıkıyı dahil et*/
main ( )
{
    printf ("Merhaba, Dünya\n");
    printf ("Bu benim ilk C programım\n");
}
```

Programın ilk satırı # işareti ile başladığı için bu satırın bir derleyici komutu olduğu anlaşılır. include komutu, derleyiciye stdio.h kütüğünü bulmasını ve derlemekte olduğu programa eklemesini söyler. Stdio.h özel bir kütüktür. Giriş ve çıkış için komutlar içerir.

Main () özel bildirimini, kendisini izleyen bloğun, program işletiminin başladığı yer olduğunu bildirir.

Printf () işletilebilir bir komuttur. Parantezler arasına yazılan herhangi bir metni ekranda yazdırır. Metin üstten (" ") işaretleri arasına yazılmalıdır.

\n karakterleri printf () tarafından özel olarak yorumlanır. Altteki satırın başına geçilmesini sağlar. <ENTER> tuşu gibi.

DEĞİŞKENLER

Bir C değişkeni, bir kutuya benzer. Kutular nesnelerin saklanması için kullanılırlar, değişik şekillerdeki nesneler, değişik şekillerdeki kutularda saklanırlar ve her bir kutuya tek bir nesne konulabilir. Değişkenlerin farklı tipleri vardır. Değişkenin tipi, içinde nasıl bir nesne saklanabileceğini belirler. En basit değişken tipi int değişkenlerdir. (Tamsayı anlamındaki integer için kullanılan bir kısaltmadır.) Bir int değişkeninde -32768 ile 32767 arasındaki herhangi bir sayı saklanabilir fakat kesir kullanılamaz.

```
78 456 9 -38 -89 0 geçerli
9.8 5/8 6.666666 geçersiz
```

DEĞİŞKENLERİN BİLDİRİLMESİ

Bir değişkenin bildirimini yapmak için, değişkenin tipi ardından da değişkenin adı yazılır. Değişken isimleri, bir harfle başlamalı ve yalnızca harfler, sayılar ve alt çizgileri (_) içermeleri gerekir. 32 karakterden fazla olmamalıdır.

ÖRNEK

```
m i sayı dur son_nokta toplam : geçerli
998 7son bu.sonuc adı? gir+bir : geçersiz
```

DEĞİŞKENE DEĞER VERİLMESİ

C'de bir değişkene bilgi yerleştirmek için eşit(=) işareti kullanılır.

```
a=10; b=100;      c=0;
```

gibi

ÖRNEK

```
# include <stdio.h>
main( )
{
  int a;
  a=10;
  printf("a'nın değeri %d\n", a);      }
```