

İskenderun Teknik Üniversitesi



Fakülte: Mühendislik ve Doğa Bilimleri *Fakültesi*

Bölüm: *Bilgisayar Mühendisliği Bölümü*

Ders: Algoritmalar ve Programlama

Dönem: 2021 – 2022 (Güz)

Öğretim üyesi: *Dr. Öğr. Üyesi Mehmet SARIGÜL*

4.

Hafta

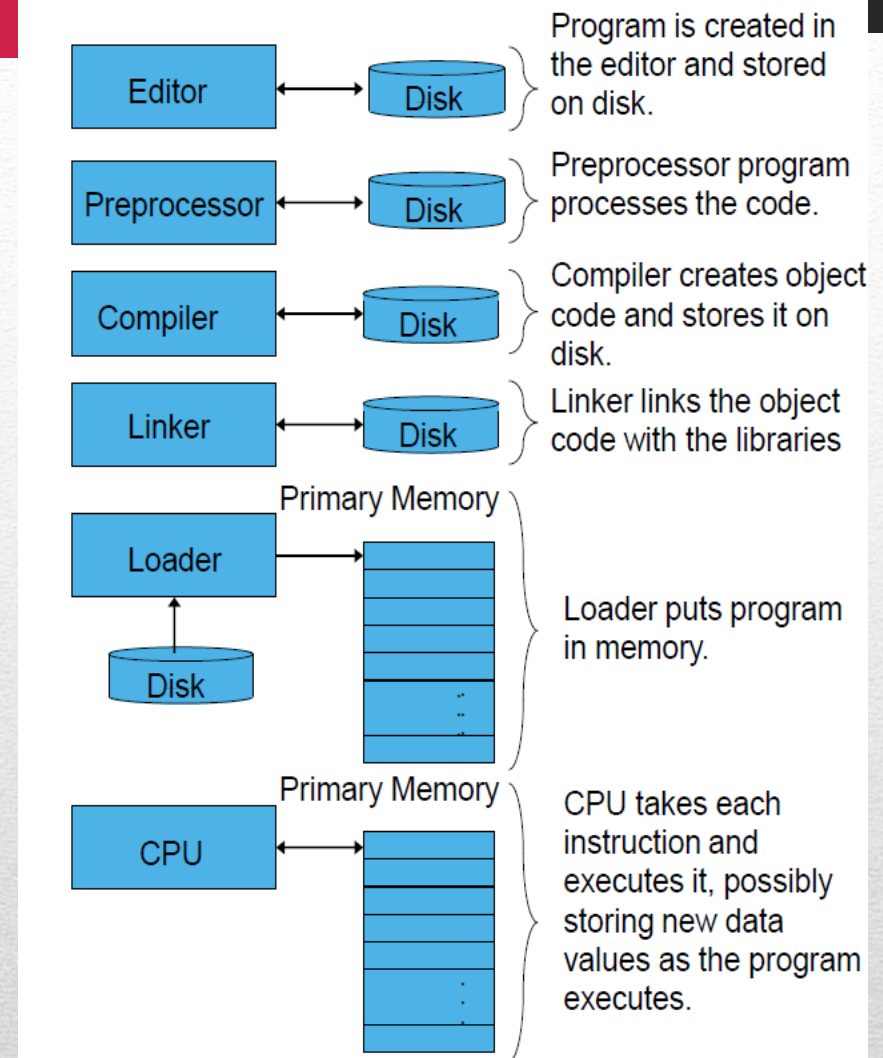
- C, 1972'de AT&T Bell Laboratuvarlarında Denis Ritchie tarafından geliştirilen üst düzey bir programlama dilidir.
- Modern işletim sistemleri yazmak için kullanılır
- Yüksek ve düşük seviyeli dillerin bazı önemli niteliklerini birleştirir.
- Programcıların sistem yazılımının yanı sıra uygulama yazılımı geliştirmesine izin verir

C programlama dili

- C++ dili, nesne yönelimli yüksek seviyeli bir programlama dilidir
- C++, New Jersey, Murray Hill'deki Bell Telephone Laboratories'de (şimdi Nokia Bell Labs olarak biliniyor) Danimarkalı Bilgisayar Bilimcisi Bjarne Stroustrup tarafından oluşturuldu, tasarlandı ve geliştirildi.
- C ++ başlangıçta "sınıflarla C" olarak biliniyordu ve 1983'te C ++ olarak yeniden adlandırıldı. ++, programlamada çeşitliliğe bir tane eklemek için kısaltmadır; bu nedenle C ++ kabaca "C'den bir yüksek" anlamına gelir.

C++ programlama dili

- 1. Düzenleme
- 2. Önışleme
- 3. Derleme
- 4. Bağlama
- 5. Yükleme
- 6. Çalıştırma



Programlama akışı



- Programcı programın performansından memnun kaldığında döngü sona erer.

Programlama Süreci

```
/* Hello World Example */
```

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

```
int main (void)
```

```
{
```

```
    printf("Hello world!\n");
```

```
    return 0;
```

```
}
```

Basit bir C programı

- `/* Hello World Example */`
- `/ * ve * /` bir yorum olduğunu belirtir
- `/ * ve * /` ile çevrili metin derleyici tarafından göz ardı edilir.
- Programı tanımlamak için kullanılır.
- Yorum Satırı: `//` derleyici oradan satırın sonuna kadar her şeyi yok sayar

Bilgi satırları

- `#include<stdio.h>`
- Önışlemci yönergeleri, görevi derlenmeden önce bir C programının metnini değiştirmek olan C ön işlemcisine talimatlar veren komutlardır.
- Bir önışlemci yönergesi, boş olmayan ilk karakteri olarak bir diez simgesiyle (#) başlar.

Önışlemci yönergeleri

- `#include` yönergesi, bir programın belirli bir kitaplığa erişmesini sağlar
- `#include <stdio.h>`, önişlemciye programda kullanılan bazı adların standart başlık dosyasında `<stdio.h>` içerisinde bulunduğunu bildirir.

Önişlemci yönergeleri

- SYNTAX:

```
#include <standard header file>
```

- Örnek:

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

```
#include <math.h>
```

-- Tüm C kodları C++ içerisinde çalışır!

#include komutu

- `int main(void)`
- C programları, biri "main" olmak üzere bir veya daha fazla fonksiyon içerir
- `int`, `main`'in bir tamsayı değeri "döndürdüğü" anlamına gelir
- `void`, işlevin hiçbir argüman almadığını gösterir.
- Kaşlı ayraçlar, `{ve}` bir bloğu gösterir. Tüm fonksiyonların gövdeleri parantez içinde yer almalıdır.

Main fonksiyonu

- SYNTAX:

```
int main(void)
{
    // fonksiyon gövdesi
}
```

main fonksiyonu

- printf işlevi, biçim dizesinin değerini görüntüler.
- Tüm satıra deyim denir. Tüm ifadeler noktalı virgül (;) ile bitmelidir.
- Kaçış karakteri (\) printf öğesinin olağan dışı bir şey yapması gerektiğini belirtir
- \n yeni satır karakteridir

printf kullanımı

- `return 0;`
- Ana işlevdeki son satır.
- Kontrolü programınızdan işletim sistemine aktarır.
- Bu durumda 0 değeri, programınızın hatasız çalıştığını gösterir.
- Sağ küme ayracı } main fonksiyonunun sonuna ulaşıldığını gösterir.

Return ifadesi

- Token: Derleyici için anlamlı olan bir programın en küçük ögesi.
- C'deki token türleri:
 - Ayrılmış kelimeler
 - Tanımlayıcılar
 - Sabitler
 - Operatörler
 - Noktalama işaretleri

Dilin elemanları

- C’da özel bir anlamı olan bir kelime
- standart kitaplıktan tanımlayıcılar ve bellek hücreleri için isimler
- küçük harfle yazılırlar
- başka amaçlar için kullanılamazlar

Ayrılmış kelimeler

auto	double	int	struct
break	else	long	switch
case	enum	register	typedef
char	extern	return	union
const	float	short	unsigned
continue	for	signed	void
default	goto	sizeof	volatile
do	if	static	while

Ayrılmış kelimeler

- Tanımlayıcılar şunlar için kullanılır:
 - Değişken isimler
 - Fonksiyon isimleri
 - Makro isimler

Tanımlayıcılar

- Bir tanımlayıcı yalnızca harflerden, rakamlardan ve alt çizgilerden oluşmalıdır.
- Bir tanımlayıcı bir rakamla başlayamaz.
- C için ayrılmış bir kelime tanımlayıcı olarak kullanılamaz.
- Bir C standart kitaplığında tanımlanan bir tanımlayıcı yeniden tanımlanmamalıdır.
- Büyük / küçük harfe duyarlı ör. Ali ve ali iki farklı tanımlayıcıdır.

Tanımlayıcılar

- Geçerli

letter_1, letter_2, inches,
CENT_PER_INCH,
Hello, variable

- Geçersiz

1Letter double int TWO*FOUR joe's

Tanımlayıcılar

- Değişken
 - değeri değişebilen bir hafıza hücresiyle ilişkili bir isim
- Değişken tanımlama
 - derleyiciye programdaki değişkenlerin adlarını ve her değişkende depolanan bilgi türünü bildiren ifadeler
 - Değişkenler kullanılmadan önce bildirilmelidir, eğer bunlar ihlal edilirse bir sözdizimi (derleme zamanı) hatası alınır
 - Her değişkenin bir adı, türü, boyutu ve değeri vardır

Değişken tanımlama

- Tamsayı için «int»
- Karakter için «char»
- Kayan noktalı sayı «float»
- Çift hassaslıkta kayan noktalı sayı «double»
- Veri türü değiştiricileri
 - signed / unsigned (int ve char için)
 - short / long

Basit veritipleri

• signed char (8 bits)	-127 to +127
• unsigned char	0 to 255
• short int (16 bits)	-32,767 to +32,767
• unsigned short int	0 to 65,535
• int (32 bits)	-2,147,483,647 to +2,147,483,647
• unsigned int	0 to 4,294,967,295
• long int (32-64 bits)	-2,147,483,647 to +2,147,483,647
• unsigned long int	0 to 4,294,967,295
• float	$\sim 10^{-37}$ to $\sim 10^{38}$
• double	$\sim 10^{-307}$ to $\sim 10^{308}$
• long double	$\sim 10^{-4931}$ to $\sim 10^{4932}$

Veritipleri

- Bir bildirim, bir veri türü adının ardından bu türden (bir veya daha fazla) değişken listesinden oluşur:
 - `char c;`
 - `int num1, num2;`
 - `float rate;`
 - `double avarage;`

Değişken tanımlama

- Bir değişken tanımlanırken değer atanabilir.
 - `char c = 'a';`
 - `int a = 220, b = 448;`
 - `float x = 0.00123`
 - `double y = 123.00`

Değişken tanımlama

- Değer atanmamış değişkenlerin anlamsız değerleri olabilir.
- Bir değişkene yeni bir değer yerleştirildiğinde, önceki değer silinir.
- Değişkenleri bellekten okumak onları değiştirmez.

Değişken tanımlama

- C, sabitler için sembolik isimler tanımlamak için bir #define yönergesi sağlar:

```
#include<stdio.h>
#define PI 3.14
```

```
int main(void)
{
    float area;
    float radius=3;
    area=PI*radius*radius;
    return 0;
}
```

Basit Makrolar

- Aritmetik operatörler
- Atama operatörü
- Mantıksal operatörler

Operatörler

Arithmetic Operator	Meaning	Example
+	addition	$5 + 2$ is 7 $5.0 + 2.0$ is 7.0
-	subtraction	$5 - 2$ is 3 $5.0 - 2.0$ is 3.0
*	multiplication	$5 * 2$ is 10 $5.0 * 2.0$ is 10.0
/	division	$5.0 / 2.0$ is 2.5 $5 / 2$ is 2
%	remainder	$5 \% 2$ is 1

Aritmetik işlemler

Operatorler	İşlemler	İşlem sıraları
()	Parentheses	Parantezler içten dışa doğru sırayla hesaplanır. Aynı seviyede parantezler soldan sağa hesaplanır.
*	Multiplication	İkincil olarak işlenir. İşlem sırası soldan sağadır.
/	Division	
%	Remainder	
+	Addition	Üçüncü olarak işlenir. Toplama ve çıkarma yapılır.
-	Subtraction	
=	Assignment	En son olarak işlenir.

Aritmetik işlemlerde sıra

Step 1. $y = 2 * 5 * 5 + 3 * 5 + 7;$ (Leftmost multiplication)

$2 * 5$ is 10

Step 2. $y = 10 * 5 + 3 * 5 + 7;$ (Leftmost multiplication)

$10 * 5$ is 50

Step 3. $y = 50 + 3 * 5 + 7;$ (Multiplication before addition)

$3 * 5$ is 15

Step 4. $y = 50 + 15 + 7;$ (Leftmost addition)

$50 + 15$ is 65

Step 5. $y = 65 + 7;$ (Last addition)

$65 + 7$ is 72

Step 6. $y = 72$ (Last operation—place 72 in y)

- değişken = ifade;

- ifade:

- operasyonlar

- değişkenler

- sabitler

- fonksiyon çağrılarını olabilir.

- Atama operatörünün önceliği, aritmetik operatörlerden daha düşüktür

$x = 5*y + (y-1)*44 ;$



Atama operatörü

- bir hesaplama sonucunun değerini bir değişkende depolayan bir talimat
- $x = y;$
- $x + 1 = 3;$ geçersiz
- Atama hafızadaki sabit bir konuma başvurmalıdır

Atama operatörleri

- Sonek Artış/Azalış
 - `int a = 3;`
 - `a++;` \backslash `a = a + 1`
- Sonek artış ifadesinin değeri, değişken artırılmadan önce belirlenir
 - `x = a ++;`
 - 1. `x = a;`
 - 2. `a = a + 1;`

Arttırma ve azaltma operatörleri

- Önek Artış/Azalış
 - `int a = 3;`
 - `++a; \\ a = a + 1`
- Etki, işleneni içeren ifade değerlendirilmeden önce gerçekleşir.
 - `x = ++ a;`
 - 1. `a = a + 1;`
 - 2. `x = a;`

Arttırma ve azaltma operatörleri

```
/** increment and decrement expressions */
#include <stdio.h>
int main(void)
{
    int a = 0 , b = 0, c = 0;
    a = ++b + ++c;
    a = b++ + c++;
    a = ++b + c++;
    a = b-- + --c;
    return 0;
}
```

Arttırma ve azaltma operatörleri

- $\text{toplam} = \text{toplam} + x;$
- kısaltılabilir
 - $\text{toplam} += x;$
- $\text{operand1} = \text{operand1} \text{ operatör } \text{operand2}$
- $\text{operand1} \text{ operatör} = \text{operand2}$

Bileşik Atama Operatörü

- Bir tamsayı ile bir tam sayı arasındaki bir aritmetik işlem her zaman bir tamsayı sonucu verir.
- Float ve float arasındaki aritmetik işlem her zaman bir float sonucu verir.
- Bir tamsayı ve bir kayan nokta arasındaki bir aritmetik işlemde, tam sayı önce float'a çevrilir ve ardından işlem gerçekleştirilir. Bu nedenle, her zaman bir kayan noktalı sonuç verir.
- Bir tam sayıya bir kayan noktalı sayı atandığında, kayan noktalı sayı bir tam sayıya indirgenir.
- Bir kayan noktalı sayıya bir tamsayı atandığında, bir kayan noktalı sayıya yükseltilir.

Tamsayı / Kayan Dönüşümler

- giriş işlemi
 - verileri bir giriş cihazından belleğe kopyalayan bir talimat
- çıktı işlemi
 - bellekte depolanan bilgileri görüntüleyen bir talimat
- giriş / çıkış işlevi
 - bir giriş veya çıkış işlemi gerçekleştiren bir C++ işlevi
- fonksiyon çağırısı
 - Bir fonksiyonu aktive etmek ya da çağırmak

Giriş ve Çıkış

- printf İşlevi, bir program çıktısı satırını görüntüler.
- fonksiyon argümanı
 - fonksiyon adından sonra parantez içine alınır
 - işlevin ihtiyaç duyduğu bilgileri sağlar
- biçim dizesi
 - printf çağrısında, çıkış satırının biçimini belirten, tırnak içine alınmış bir karakter dizisi
- printf(“<format dizesi>, <değişkenler listesi>)

printf

- baskı listesi
 - printf çağrısında, değerleri görüntülenen değişkenler veya ifadeler
- Yer tutucu
 - çıktı değerinin nerede görüntüleneceğini gösteren bir biçim dizesinde % ile başlayan bir sembol

`printf("That equals %f kilometers. \n", kms);`

- Biçim dizesindeki yer tutucular
- %c: argüman tek bir karakter olarak alınır
- %d: argüman bir tamsayı olarak alınır
- %f: argüman kayan nokta olarak alınır (kayan nokta veya çift)
- %s: argüman bir dize olarak alınır

printf

- Standart giriş aygıtındaki (genellikle klavye) verileri bir değişkene kopyalar.
- `scanf("<format dizesi">, <değişkenlerin adresi>)`

`scanf("%d", &number);`

scanf

```
#include <stdio.h>
int main()
{
    int a,b;
    scanf("%d %d",&a,&b);
    printf("%d",a+b);
    return 0;
}
```


- Girilen fahrenheit derecesini santigrat dereceye dönüştürmek için bir C programı yazın.

Egzersiz



- Verilen 5 basamaklı bir sayıyı ters çevirmek için bir C programı yazın.

Ödev





Ders Sonu

