

BIL 121 Yapısal Programlama

Giriş

-
- ▶ Dr.Ayşe Betül Oktay
 - ▶ Ofis: F 126
 - ▶ abetul.oktay@medeniyet.edu.tr



Ders Saatleri

- ▶ Teorik Ders: Çarşamba/Cuma 10:30-12:15 K602
- ▶ Lablar:
 - ▶ Cuma 13:30 - 15:00 numarası tek olanlar
 - ▶ Cuma 15:30 - 17:00 numarası çift olanlar
- ▶ Derse ve lablara katılım zorunludur.
 - ▶ Labların asgari %80'ine veya dersin %70'ine katılmadığınızda dersten kalırsınız.
 - ▶ Herhangi bir sebeple katılmadığınız lablardan sıfır alırsınız.
 - ▶ Çok ciddi bir mazeretiniz varsa bildiriniz.
 - ▶ Cuma laba gelmeden önce öğrenci numaranızı öğreniniz.



Dersin Muhtevası

- ▶ Bu derste bilgisayar programlama paradigmasını (bir şeyin nasıl üretileceği konusunda örnek, model) öğreneceksiniz:
 - ▶ Hesaplama ve algoritma tabanlı düşünme
 - ▶ Hesaplama Karmaşıklığı
 - ▶ C programlamanın syntax ve semantic yapısını
 - ▶ Basit bilimsel ve mühendislik problemlerine uygulamaları
- ▶ Bu derste C programlama dilini (programlama aracını) kullanarak size programlama becerisi kazandırmayı amaçlamaktadır.
 - ▶ Derste öğreneceğiniz bir çok konu ve kavram diğer programlama dilleri içinde geçerlidir.



Bu Derse Nasıl Çalışmalıyım?

- ▶ **1.5 saatlik haftalık derslerde**
 - ▶ Konu materyalini ve basit örneklerle uygulanmasını öğrenip/göreceksiniz.
 - ▶ Sınıfta bol bol soru sorun
 - ▶ Ve Not alın
- ▶ **Evde**
 - ▶ Konu materyalini tekrar çalışıp sınıfta gösterilen örnekleri kendi kendinize yeniden yapın
 - ▶ Ödevleri kendi kendinize yapın
- ▶ **1.5 saatlik haftalık lablarda**
 - ▶ Bir önceki dersin konusunun pratiğini farklı örnekler üzerinde yapacaksınız.



Ders notları

- ▶ Slaytlar Google classrooma yüklenecektir.
- ▶ Bir gmail hesabınızın olması gerekmektedir.
- ▶ <http://classroom.google.com/>
- ▶ Adresine girip **kgr6si** kodunu girerek Google classrooma üye olunuz.

Kitap

- ▶ İsteddiğiniz kitabı kullanabilirsiniz.
- ▶ İnternet ve youtube kaynaklar
- ▶ Deitel - C How to program



Varsayımlar

- ▶ Sıfır programlama bilgisine sahipsiniz.
- ▶ Programlamaya çok ilgilisiniz ve derslerde uyumayacaksınız.
 - ▶ Yada en azından horlamayacaksınız.
- ▶ Derste telefonlarınızı kullanmayacaksınız.
- ▶ Çok çalışkansınız; ödev ve projelerinizi zamanında yapacaksınız.



Notlar

- ▶ Lütfen ders ve lablarda dakik olun
 - ▶ zamanında gelip, zamanında çıkın
- ▶ Derste herhangi bir zamanda soru sorabilirsiniz
- ▶ Ödev ve projeler bireysel çalışmanız içindir.



Beklenti: Akademik Doğruluk

- ▶ Aksi bildirilmediği sürece tüm proje/ödev/lab/sınavları hocalarınızdan başkasından yardım almadan yapmalısınız ve yapmak zorundasınız.
- ▶ Arkadaşınızdan, internetten vs aldığınız kodlar kopya olarak nitelendirilecektir.
- ▶ Kopya veya benzer ödevler -100 puan alır.



Derecelendirme

Lablar	%10 (1 en düşük lab dahil değil)
Ödev ve projeler	%25 (1 en düşük ödev dahil değil)
Vize	%25
Final	%40



Derecelendirme

Başarı Derecesi	Başarı Notu	Puanı	Puanı
Pekiyi	AA	4,00	90-100
İyi-pekiyi	BA	3,50	85-89
İyi	BB	3,00	80-84
Orta-iyi	CB	2,50	70-79
Orta	CC	2,00	60-69
Geçer-orta	DC	1,50	55-59
Geçer	DD	1,00	50-54
Zayıf	FF	0,00	0-49



GİRİŞ



Bu Hafta



- ▶ Giriş:
 - ▶ bilgisayar nedir?
 - ▶ programlama dilleri,
 - ▶ syntax, semantic hataları
- ▶ C programını yazma, derleme (compile) ve çalıştırma (execution)



Bilgisayar



- ▶ Bilgisayarlar kalıtsal olarak zeki değillerden ve çok güzel bir şekilde **aptal** olabilirler.
- ▶ Çok kısıtlı işlemler yapabilen basit makinelerdir.
- ▶ Kendilerine verilen komutları takip ederler.
- ▶ 2019 itibariyle halen daha ne yaptıklarını anlayabilmiş değillerdir.



Bilgisayar

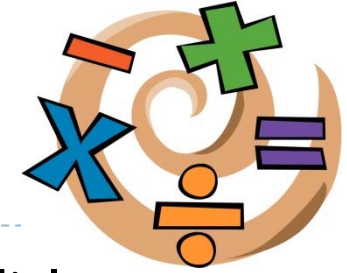


- ▶ Günümüzde kullandığımız bilgisayarlar dijitaldirler.
 - ▶ Bir dijit sadece 1 veya 0 olabilir
 - ▶ Bilgisayarlar sadece 1ler ve 0larla ifade edilen bilgiyi anlarlar
 - ▶ 1 digit 1-bitlik bilgi ifade eder

- ▶ Gerçek dünyadaki mefhumu bitlerle nasıl ifade ederiz?



Bilgisayar



- ▶ Bilgisayarlar sadece belirli basit işlemleri yapabilmektedirler.
 - ▶ Aritmetik işlemler
 - ▶ toplama,
 - ▶ çıkarma,
 - ▶ bölme,
 - ▶ çarpma
 - ▶ Ve bazı mantıksal işlemler.

- ▶ Oyun programları gibi karmaşık görevleri bu basit işlemleri kullanarak nasıl başarabiliriz?



Bir Problemi Nasıl Çözeriz?



- ▶ Polya'nın problem çözme tekniği (George Polya 1945)
 - ▶ Problemin anlaşılması
 - ▶ Bir planın oluşturulması
 - ▶ Planın hayata geçirilmesi
 - ▶ Tekrar Geriye Dönme (planın revize edilmesi)
- ▶ Bir bilgisayarı kullanarak bir problemi nasıl çözeriz?



Bilgisayar Mimarisi

Bilgisayarlar 0 ve 1lerle çalışırlar

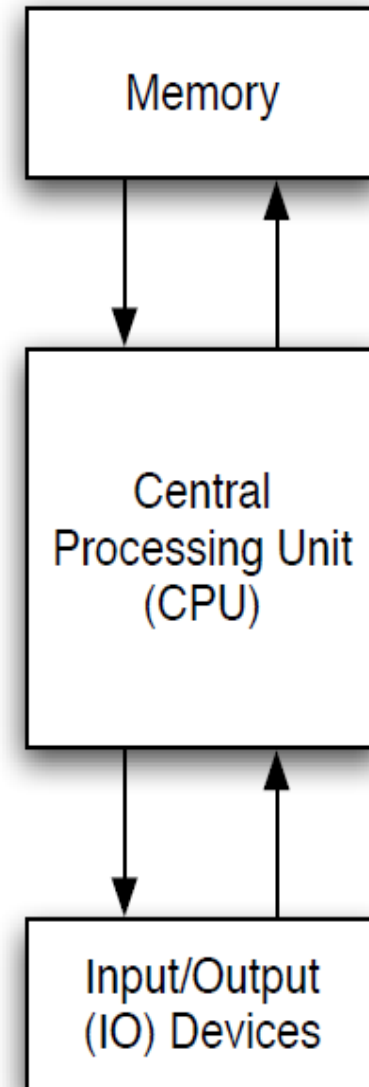
İki tür bilgi bilgisayar hafızasında saklanır.

- ▶ Komut(instruction) CPU tarafından çalıştırılır.
- ▶ Veri(Data) CPU tarafından manipüle edilir

Bir program hafızada saklı olan bir komut kümesidir.

Yazılım(Programlar) donanıma (CPU, I/O aygıtları, ve hafıza) erişim ve kontrol için kullanılırlar.

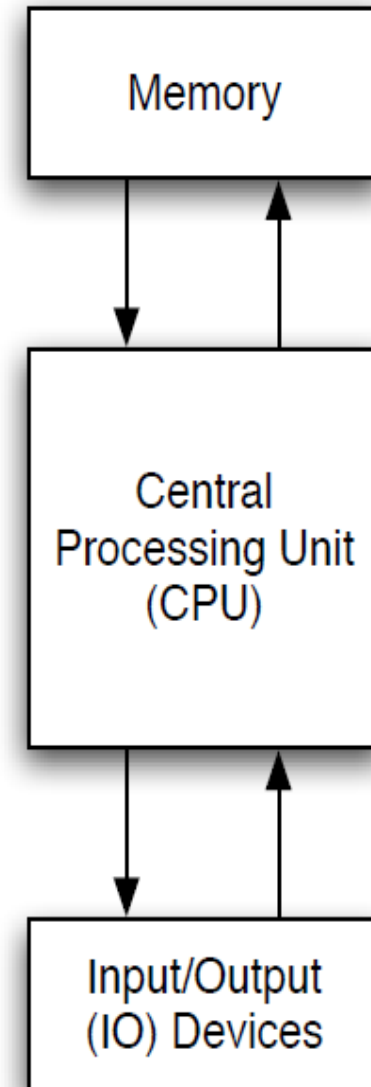
- ▶ Bu işlemler yine kendisinde yazılım olan işletim sistemi üzerinden yapılır.
 - ▶ İşletim sistemi bilgisayar üzerinde devamlı çalışmaktadır.



Bilgisayar Mimarisi

$z = x + y$ için CPU komutları

- ▶ Hafızada x yerini oku (read)
 - ▶ Hafızada y yerini oku
 - ▶ Ekle
 - ▶ Sonucu hafızada z yerine yaz (write)
- ▶ Bir program sıralı bazı işlemler gerçekleştiren bir komut kümesidir:
- ▶ Tipik olarak giriş datası alır.
 - ▶ Bu datayla veya bu data üzerinde bazı işlemler yapar (algoritma).
 - ▶ Bir çıkış datası oluşturur.



Bir problemi programla nasıl çözeriz?

- ▶ **Polya'nın problem çözme tekniği**
 - ▶ Problemin anlaşılması
 - ▶ Bir planın oluşturulması
 - ▶ Planın hayata geçirilmesi
 - ▶ Tekrar Geriye Dönme (planın revize edilmesi)
- ▶ Problemin çözümüne dönük program yazmadan önce
- ▶ Şunları belirlememiz lazım
 - ▶ Programın inputu,
 - ▶ Algoritma
 - ▶ Bir problemin çözümü için belirlenmiş adımlar
- ▶ Programın outputu(çıktısı)



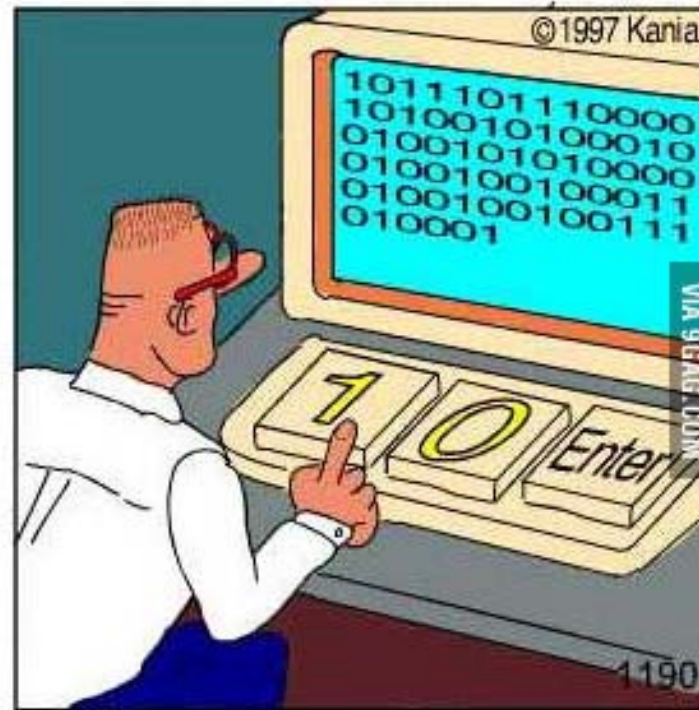


Programlama Dilleri

- ▶ Bir bilgisayar (CPU) sadece kendi dilinde (CPU komut kümesi) yazılanı anlar.
 - ▶ Bu dil **Makina Dili (machine language)** olarak adlandırılır.
 - ▶ **11000011 1000010100100000**
 - ▶ **Komut** **Adres**
- ▶ Her CPU'nun kendine ait komut kümesi vardır: Bir CPU için yazılan komutlar diğer CPUda çalışmayabilir.
- ▶ Makine dilinde yazılan bir program çalıştırılabilir (**executable** yada **binary code**) olarak adlandırılır.



Makine Diliyle Program Yazma



Real programmers code in binary.



[illegible]

- Bilgisayar programlarının daha insani bir dil kullanılarak yazılması
- Sonra yazılan bu programın derleyici(compiler) kullanılarak makine diline çevrilmesi
- Assembly Dili
 - Herbir makine komutu için bir komut içerir
 - 00000101 yerine DEC B
 - Assembly programını makine diline çevirmek için Assembler kullanılır
- Daha yüksek seviyeli dilller: C, C++, Java, Object C, C#, Python, Ruby, vs.



Makine Diline Çevirim

- Yorumlayıcı (interpreter) yorumlanabilir bir dilde (interpreted-language) yazılmış herhangi bir programı çalıştırabilecek bir programdır.
- Derleyici (compiler) yüksek seviyeli bir dilde yazılmış bir programı makine diline çevirir.



Programın (bir kodun) Yapısı

- **Sözdizimi (Syntax)** dilbilgisi yapısıyla ilgilidir. Doğru sözdizimi grameri düzgün cümle veya cümlecikleri ifade eder.
 - Mesela, tahtae: Türkçe de olmadığı için ya tahta olmalı yada başka bir şey
 - C de, herbir cümleciğin noktalı virgülle ayrılması.
- **Mantıksal yapı (Semantic)** anlamla ilgilidir. Verilen bir cümlecik mantiki olarak bir anlam ifade ediyor mu etmiyor mu.
 - 10/"print" mantiki olarak yapılamaz, çünkü bir sayıyı metine bölemezsiniz.



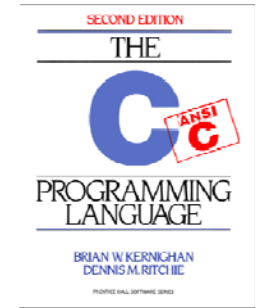
Syntax ve Semantic hataları



- ▶ **Syntax hataları** yazdığınız programın programlama dilinin kurallarından bir veya birkaçını ihlal ettiği durumlarda oluşur.
 - ▶ Programı derleyemez ve çalıştıramazsınız.
- ▶ **Semantic hataları** genelde programın çalışması esnasında ortaya çıkan hatalardır.



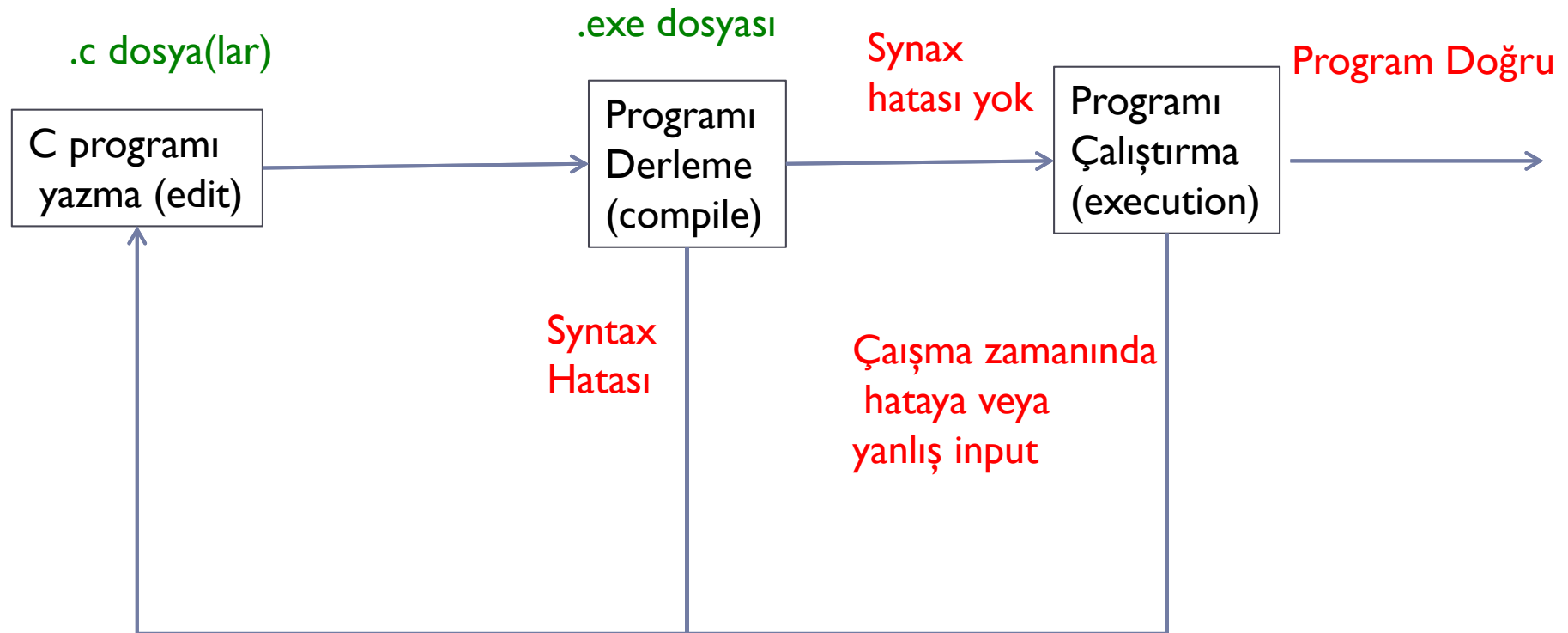
C Programlama Dili



- Dennis Ritchie – AT&T Bell Laboratories – 1972
- Unix işletim sistemi C de yazıldı.
- Yeni sistemlerde ve mimarilerde halen yaygın bi şekilde kullanılmaktadır.
- Verimli ve performansı yüksektir.
- Düşük seviyeli erişim sağlar.
- Kalıtımsal olarak güvenli değildir.



C de Program Yazma



İlk Programı Yazma

- Aşağıdaki metni bilgisayar ekranına yazan program:
 - Hello Medeniyet!
- Program yazmak için Editöre ihtiyacımız var.
 - Linux: Vim, Emacs, Pico, Nano, Gedit, vs.
 - Windows: Notepad, WordPad, Geany, Eclipse
 - Lablarda Geany kullanacağız.
 - Cep telefonu ya da bilgisayarınızdan internet varsa repl.it de kullanabilirsiniz.



İlk Programımızı Yazma

```
/* hello.c  
* prints Hello Medeniyet!  
* @adaskin  
*/
```

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

```
int main (){
```

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

```
    return 0;
```

```
}
```



İlk Programımızı Yazma

```
/* hello.c
```

```
* prints Hello Medeniyet!
```

```
* @adaskin
```

```
*/
```

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

```
int main () {
```

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

```
return 0;
```

```
}
```

Yorumlar: `/*` ... `*/`
yorumlar derleyici tarafından göz ardı edilir.

`#include <stdio.h>` (önişleme)
preprocessor komutu. Derleme işleminden önce derlenecek kodda `stdio.h` dosyasının da içerilmesini sağlar.

`int main()` programın çalışmaya başladığı ana fonksiyon.

`printf(...)` C de kullanılabilecek fonksiyonlardan birisi "Hello, Medeniyet!" yazısını ekrana yazdırır.

`return 0;` main fonksiyonunu sonlandırır ve 0 değerini döndürür.



İlk Programımızı Yazma

- Yazmış olduğumuz dosyayı hello.c şeklinde kaydedelim.
- “.c” C-programlama dosyası olduğunu belirtmektedir.



Programı Derleme

- hello.c dosyasını binary dosyaya çevirmek için derleyiciye ihtiyacımız var.
 - Linux: gcc, build-essential paketinin bir parçası
 - Windows: visual studio compiler içeriyor.
- Linux terminalde
 - > gcc hello.c -o hello
- Windows komut satırında
 - > gcc hello.c -o hello.exe
 - Gcc yüklü değilse
 - > cl hello.c /out:hello.exe
- Bunlar çalıştırılabilir dosya üretmektedir:
 - Linux de hello,
 - Windows da hello.exe



Programı Çalıştırma

- Windows'da hello.exe
- Linux'de ./hello
- “Hello Medeniyet!” yazısını ekranda görmeniz gerekiyor.



Farklı Örnekler

```
/* hello2.c
```

```
* @adaskin
```

```
*/
```

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

```
int main (){
```

```
    printf("Hello");
```

```
    printf(" Medeniyet!");
```

```
    printf("\n");
```

```
    return 0;
```

```
}
```



Farklı Örnekler

```
/* hello2.c
```

```
* @adaskin
```

```
*/
```

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

```
int main (){
```

```
    printf("Hello");
```

```
    printf(" Medeniyet!\n");
```

```
    return 0;
```

```
}
```

