



# **Bölüm 1: Programlama Dilleri**

## **Mikroişlemciler**



# Yazılım

- Bilgisayar donanımını düzgün, koordineli ve istenen şekilde çalıştırır.
- İşletim sistemi,
  - bilgisayarın temel yazılımıdır.
  - diğer yazılımların çalışmasını denetler.
- Programlama dilleri,
  - yazılım geliştirmek için icat edilmiştir.
  - farklı alanlara yönelik çeşitli diller mevcuttur.
- Programlama dillerinin evrimi,
  - bilgisayar teknolojilerindeki önemli gelişmeleri yansıtır.



# Programlama Dilleri

- Makine Dilleri
- Çevirici Diller
- Yüksek Seviyeli Diller
  - Derlenen
  - Yorumlanan



# Makine Dilleri

- 0 ve 1'lerden oluşan en alt seviyeli dil.
- Tamamen işlemci ve donanıma bağımlıdır.
- İlk bilgisayar ENIAC, tamamen donanıma bağlı bir makine dili kullanmıştır.
- Günümüzde,
  - her bilgisayar için özel bir makine dili bulunmaktadır.
  - kullanımının pratik olmaması nedeniyle doğrudan kullanılmaz.
- Çevirici diller,
  - makine diline olan bağımlılığı azaltır.
  - daha insan dostu bir programlama deneyimi sağlar.



# Makine Dili Tasarımı

- Komutlar 8 bitlidir.
- İlk 4 bit, işlemi belirtir.
  - 0000: Halt (programı durdur)
  - 0001: Toplama
  - 0010: Çıkarma
  - 0011: AND (mantıksal ve)
  - 0100: OR (mantıksal veya)
- Sonraki 2 bit, hedef yazmacı belirtir.
- Sonraki 2 bit, kaynak yazmacı belirtir.



# Örnek Kod

; Program Kodu:

0001 0001 ; R0 ve R1'i topla, sonucu R1'e kaydet

0011 1011 ; R2 ve R3'ü AND'le, sonucu R3'e kaydet

0100 0110 ; R1 ve R2'yi OR'le, sonucu R2'ye kaydet

0000 0000 ; Halt (programı durdur)



# Çevirici Dilleri

- Farklı işlemci mimarilerine ve komut setlerine uygun olarak tasarlanır.
- Motorola, Intel, PIC gibi platformlar için özel çevirici dilleri bulunur.
- Mikrodenetleyiciler için özel çevirici dilleri,
  - gömülü sistemlerde ve
  - elektronik uygulamalarda sıkça kullanılır.
- Çevirici dilleri,
  - işlemci platformları arasında uyumu sağlar ve
  - donanım özelliklerine en iyi şekilde uyumlu olmayı amaçlar.



# x86 Assembly

```
section .data
    hello db 'Merhaba, Dünya!', 0
section .text
    global _start
_start:
    mov eax, 4           ; sys_write'in sistem çağrısı numarası
    mov ebx, 1           ; stdout'un dosya tanıtıcısı
    mov ecx, hello      ; yazılacak stringin adresi
    mov edx, 15         ; yazılacak karakter sayısı
    int 0x80            ; sistem çağrısını yap
    mov eax, 1          ; sys_exit'in sistem çağrısı numarası
    xor ebx, ebx        ; hata kodu, burada 0 (başarı) olarak ayarlandı
    int 0x80            ; sistem çağrısını yap
```





# x86 Assembly (DOS)

```
section .data
```

```
    hello db 'Merhaba, DOS!', 0
```

```
section .text
```

```
    org 100h                ; Programın başlangıç adresi
```

```
    mov ah, 9               ; DOS'un 9 numaralı hizmet çağrısı (print string)
```

```
    mov dx, hello           ; yazılacak stringin adresi
```

```
    int 21h                 ; DOS hizmet çağrısını yap
```

```
    mov ah, 4Ch             ; DOS'un 4Ch hizmet çağrısı (programı sonlandır)
```

```
    int 21h                 ; DOS hizmet çağrısını yap
```



# x86 Assembly (Windows)

```
section .data
    hello db 'Merhaba, Windows!', 0
section .text
    global _start
_start:
    push 0           ; uType: MB_OK
    push hello       ; lpCaption
    push hello       ; lpText
    push 0           ; hWnd
    call MessageBoxA
    push 0           ; ExitCode: 0
    call ExitProcess
extern MessageBoxA ; MessageBoxA fonksiyonunu çağıran external fonksiyon tanımı
extern ExitProcess
```



# x86 Assembly (Linux)

```
section .data
    hello db 'Merhaba, Linux!', 0
section .text
    global _start
_start:
    mov eax, 4           ; sys_write'in sistem çağrısı numarası
    mov ebx, 1           ; stdout'un dosya tanıtıcısı
    mov ecx, hello      ; yazılacak stringin adresi
    mov edx, 15          ; yazılacak karakter sayısı
    int 0x80             ; sistem çağrısını yap
    mov eax, 1           ; sys_exit'in sistem çağrısı numarası
    xor ebx, ebx         ; hata kodu, burada 0 (başarı) olarak ayarlandı
    int 0x80             ; sistem çağrısını yap
```



# Amiga 68000 Assembly

```
SECTION mysection,DATA_C
hello DC.B 'Merhaba, Amiga!',0
SECTION mysection,CODE_C
START:
    MOVE.L #hello,A1      ; yazılacak stringin adresini A1'e yükle
    MOVE.L #14,A6         ; dos.library'nin başlatıcı adresini A6'ya yükle
    JSR (A6)              ; dos.library'nin başlatıcı adresini çağır
    MOVE.L D0,ERR         ; hizmet çağrısının sonucunu ERR etiketine kopyala
    MOVE.L #0,D0          ; hata kodunu 0 (başarı) olarak ayarla
    MOVE.L #20,A6         ; dos.library'nin başlatıcı adresini A6'ya yükle
    JSR (A6)              ; dos.library'nin başlatıcı adresini çağır
    BRA.S END
ERR    DC.L    0          ; Hizmet çağrısının sonucunu saklamak için değişken
END    STOP    #$2700    ; Programı sonlandır
```



# ARM Assembly

```
section .data
    hello db 'Merhaba, ARM!', 0
section .text
    global _start
_start:
    mov r7, 4           ; sys_write'in sistem çağrısı numarası
    mov r0, 1           ; stdout'un dosya tanıtıcısı
    ldr r1, =hello      ; yazılacak stringin adresi
    ldr r2, =15         ; yazılacak karakter sayısı
    swi 0x0             ; sistem çağrısını yap
    mov r7, 1           ; sys_exit'in sistem çağrısı numarası
    mov r0, 0           ; hata kodu, burada 0 (başarı) olarak ayarlandı
    swi 0x0             ; sistem çağrısını yap
```



# PowerPC (PPC) Assembly

```
section .data
    hello db 'Merhaba, PowerPC!', 0
section .text
    global _start
_start:
    li r0, 1           ; stdout'un dosya tanıtıcısı
    li r3, 15          ; yazılacak karakter sayısı
    li r4, 4           ; sys_write'in sistem çağrısı numarası
    mr r5, r1          ; yazılacak stringin adresi
    sc                 ; sistem çağrısını yap
    li r0, 0           ; sys_exit'in sistem çağrısı numarası
    li r3, 0           ; hata kodu, burada 0 (başarı) olarak ayarlandı
    li r4, 1           ; sys_exit'in sistem çağrısı numarası
    sc                 ; sistem çağrısını yap
```



# Yüksek Seviyeli Diller

- Derlenen Diller
  - Kaynak kod makine diline çevrilir.
  - Derleme sürecinde hatalar tespit edilir.
  - Performans avantajı sağlar.
  - Derleme aşamaları: derleme (*compiler*) ve bağlama (*linker*)
- Yorumlanan Diller
  - Kaynak kod çalışma anında yorumlanır.
  - Hızlı geliştirme ve test imkanı sağlar.
  - Kaynak kod platformdan bağımsızdır.



# Derlenen Diller

- Kaynak kod derleyici tarafından incelenir.
- Derlenmiş kod daha hızlı çalışır.
- Derleme aşamasında hatalar tespit edilir.
- **C ve C++:** sistem programlaması ve yüksek performanslı uygulamalar
- **Rust:** güvenli ve performanslı sistem programlaması
- **Fortran:** bilimsel hesaplamalar için kullanılır



# C



```
#include <stdio.h>
int main() {
    int number;
    printf("Bir sayi girin: ");
    scanf("%d", &number);
    for (int i = 1; i <= number; i++) {
        if (i % 2 == 0) {
            printf("%d sayisi bir cift sayidir.\n", i);
        } else {
            printf("%d sayisi bir tek sayidir.\n", i);
        }
    }
    return 0;
}
```



# C++

```
#include <iostream>
int main() {
    int number;
    std::cout << "Bir sayi girin: ";
    std::cin >> number;
    for (int i = 1; i <= number; i++) {
        if (i % 2 == 0) {
            std::cout << i << " sayisi bir cift sayidir." << std::endl;
        } else {
            std::cout << i << " sayisi bir tek sayidir." << std::endl;
        }
    }
    return 0;
}
1/20/2023
```

# C#



```
using System;
class Program {
    static void Main() {
        Console.Write("Bir sayi girin: ");
        int number = int.Parse(Console.ReadLine());
        for (int i = 1; i <= number; i++)
        {
            if (i % 2 == 0) {
                Console.WriteLine($"{i} sayisi bir cift sayidir.");
            } else {
                Console.WriteLine($"{i} sayisi bir tek sayidir.");
            }
        }
    }
}
```



# Java

```
import java.util.Scanner;
public class Main {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Bir sayi girin: ");
        int number = scanner.nextInt();
        for (int i = 1; i <= number; i++) {
            if (i % 2 == 0) {
                System.out.println(i + " sayisi bir cift sayidir.");
            } else {
                System.out.println(i + " sayisi bir tek sayidir.");
            }
        }
    }
}
```



# Go

```
package main
import "fmt"
func main() {
    var number int
    fmt.Print("Bir sayi girin: ")
    fmt.Scan(&number)
    for i := 1; i <= number; i++ {
        if i%2 == 0 {
            fmt.Printf("%d sayisi bir cift sayidir.\n", i)
        } else {
            fmt.Printf("%d sayisi bir tek sayidir.\n", i)
        }
    }
}
```



# Rust

```
use std::io;
fn main() {
    let mut input = String::new();
    println!("Bir sayi girin:");
    io::stdin().read_line(&mut input).expect("okuma hatasi");
    let number: i32 = input.trim().parse().expect("cevirme hatasi");
    for i in 1..=number {
        if i % 2 == 0 {
            println!("{}", sayisi bir cift sayidir.", i);
        } else {
            println!("{}", sayisi bir tek sayidir.", i);
        }
    }
}
```



# Fortran

```
program Main
  implicit none
  integer :: number, i
  write(*,*) 'Bir sayi girin:'
  read(*,*) number
  do i = 1, number
    if (mod(i, 2) == 0) then
      write(*,*) i, ' sayisi bir cift sayidir.'
    else
      write(*,*) i, ' sayisi bir tek sayidir.'
    end if
  end do
end program Main
```



# Ada

```
with Ada.Text_IO;
procedure Main is
    Number : Integer;
begin
    Ada.Text_IO.Put_Line("Bir sayi girin:");
    Ada.Text_IO.Get(Number);
    for I in 1 .. Number loop
        if I mod 2 = 0 then
            Ada.Text_IO.Put(I);
            Ada.Text_IO.Put(" sayisi bir cift sayidir.");
        else
            Ada.Text_IO.Put(I);
            Ada.Text_IO.Put(" sayisi bir tek sayidir.");
        end if;
    end loop;
end Main;
```





# Yorumlanan Diller

- Kaynak kodun çalışma anında yorumlandığı dillerdir.
- Hızlı geliştirme ve test imkanları sunar.
- **Python**: genel amaçlı, yüksek seviyeli bir dil.
- **Javascript**: web geliştirme ve tarayıcılar için kullanılır.
- **Ruby**: kullanım kolaylığı ve esneklik sunar.



# Python

```
# Kullanıcıdan input alma
number = int(input("Bir sayi girin: "))

# For döngüsü ile 1'den kendisine kadar sayıları kontrol etme
for i in range(1, number + 1):
    if i % 2 == 0:
        print(f"{i} sayisi bir cift sayidir.")
    else:
        print(f"{i} sayisi bir tek sayidir.")
```



# Javascript

```
// Kullanıcıdan input alma
let number = prompt("Bir sayi girin:");

// For ile 1'den kendisine kadar sayıları kontrol etme
for (let i = 1; i <= number; i++) {
    if (i % 2 === 0) {
        console.log(i + " sayisi bir cift sayidir.");
    } else {
        console.log(i + " sayisi bir tek sayidir.");
    }
}
```



# Ruby

```
# Kullanıcıdan input alma
print "Bir sayi girin: "
number = gets.chomp.to_i
```

```
# For ile 1'den kendisine kadar olan sayıları kontrol etme
for i in 1..number do
  if i % 2 == 0
    puts "#{i} sayisi bir cift sayidir."
  else
    puts "#{i} sayisi bir tek sayidir."
  end
end
```

end



# PHP

```
<?php
// Kullanıcıdan input alma
echo "Bir sayi girin: ";
$number = trim(fgets(STDIN));

// For ile 1'den kendisine kadar olan sayıları kontrol etme
for ($i = 1; $i <= $number; $i++) {
    if ($i % 2 == 0) {
        echo $i . " sayisi bir cift sayidir.\n";
    } else {
        echo $i . " sayisi bir tek sayidir.\n";
    }
}
?>
```



# Perl

```
# Kullanıcıdan input alma
print "Bir sayi girin: ";
my $number = <STDIN>;
chomp $number;
```

```
# For ile 1'den kendisine kadar olan sayıları kontrol etme
for my $i (1..$number) {
    if ($i % 2 == 0) {
        print "$i sayisi bir cift sayidir.\n";
    } else {
        print "$i sayisi bir tek sayidir.\n";
    }
}
```



# MATLAB

```
% Kullanıcıdan input alma
number = input('Bir sayi girin: ');

% For ile 1'den kendisine kadar olan sayıları kontrol etme
for i = 1:number
    if mod(i, 2) == 0
        fprintf('%d sayisi bir cift sayidir.\n', i);
    else
        fprintf('%d sayisi bir tek sayidir.\n', i);
    end
end
```



SON