Bölüm 1.1: Temel Kavramlar

Bilgisayar ve Programlamaya Giriş

Bilgisayarın çalışma prensibi | İkili sistem | Veri temsili | Bilgisayar türleri | Bilgi teknolojileri

Sercan KÜLCÜ · Giresun Üniversitesi

© Sercan Külcü. Tüm hakları saklıdır. · Sunumun etkileşimli HTML sürümü: https://sercankulcu.github.io/files/computer/slides/Bolum_01_01_Temel_Kavramlar.html

GİRİŞ

Öğrenme Çıktıları

Çalışma prensibi

Giriş, işlem, çıkış ve depolama aşamalarını açıklamak.

Sayısal temsil

Bit, bayt, ikili sayı sistemi ve mantıksal işlemlerin rolünü yorumlamak.

Bilgisayar türleri

Bilgisayarları kullanım amacı ve kapasitesine göre karşılaştırmak.

Bilgi teknolojileri

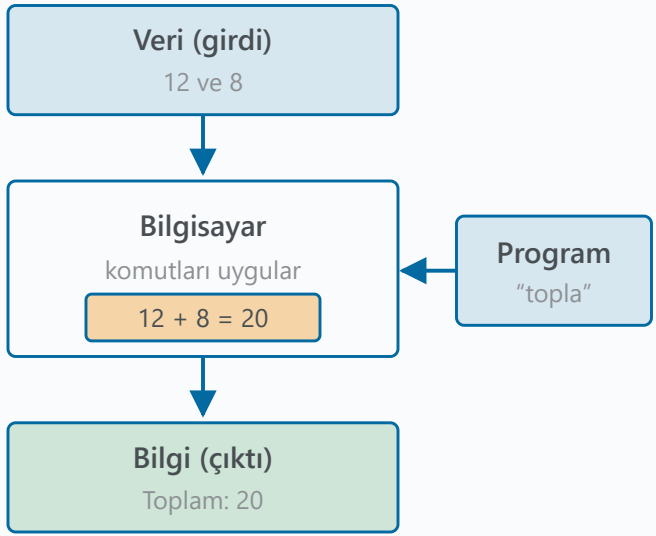
Verinin bilgiye dönüşümünü ve BT'nin günlük yaşamdaki rolünü açıklamak.

ÇALIŞMA PRENSİBİ

Bilgisayar Nedir?

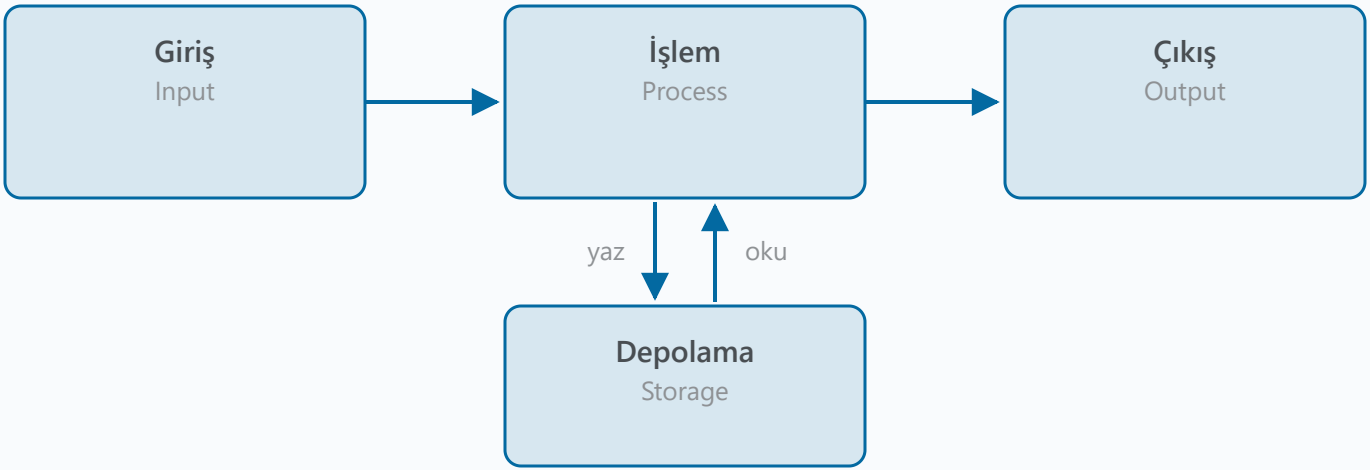
Bilgisayar, aldığı verileri programlardaki komutlara göre işleyen elektronik bir sistemdir.

- **Veri**: işlemden önceki ham değer. **Bilgi**: yorumlanabilir sonuç.
- Kullanıcıdan veya başka sistemlerden veri alır.
- Veriyi işler ve anlamlı sonuç üretir.
- Sonuçları sunar veya daha sonra kullanmak üzere saklar.



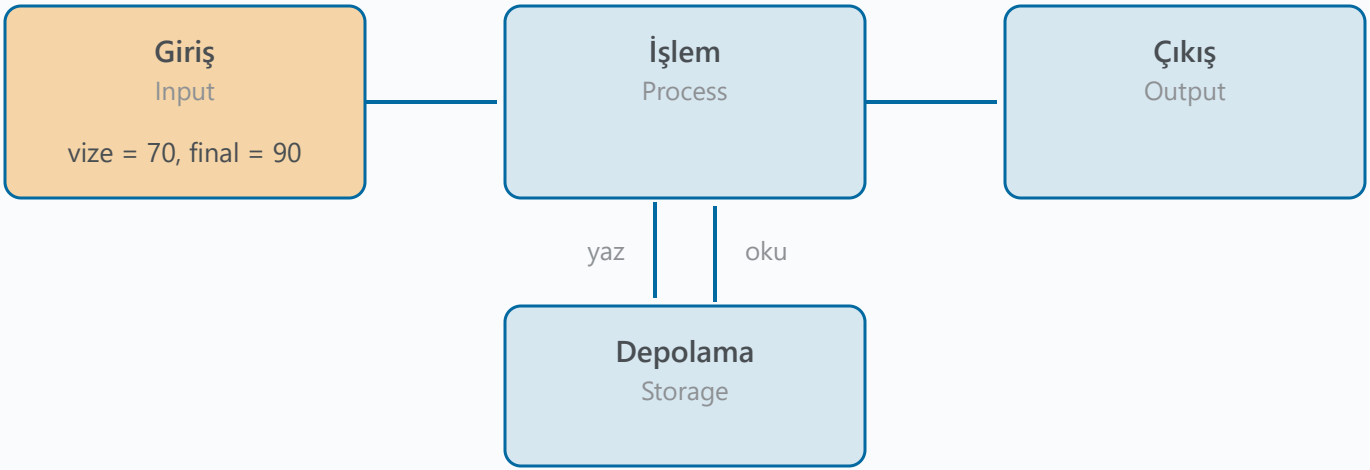
ÇALIŞMA PRENSİBİ · ADIM ADIM

Temel Çalışma Döngüsü



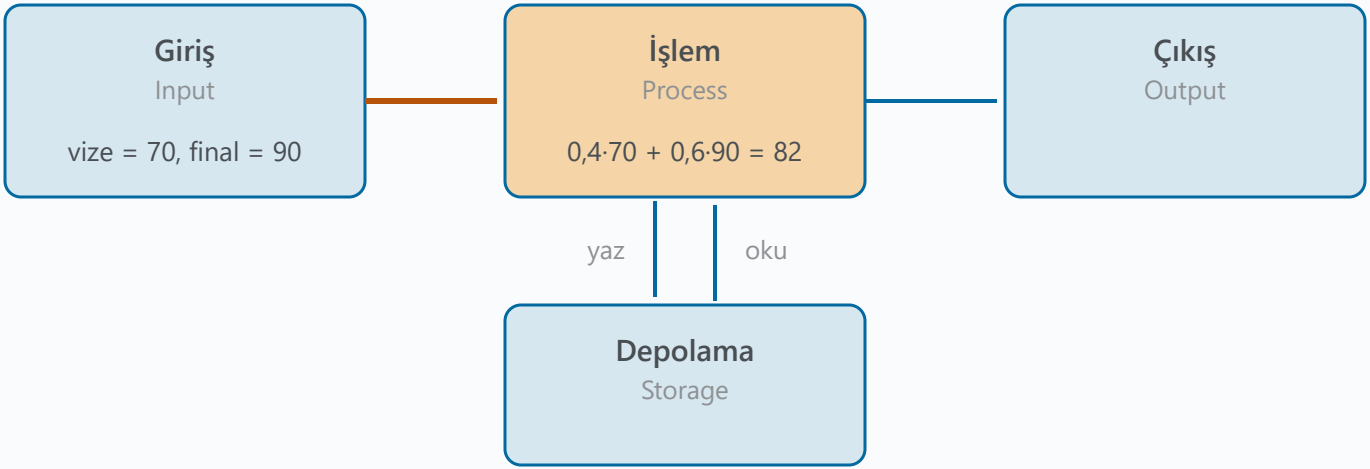
Bilgisayar her işte aynı dört aşamayı kullanır. Örnek: bir öğrenci bilgi sisteminde ortalama hesabı.

Temel Çalışma Döngüsü



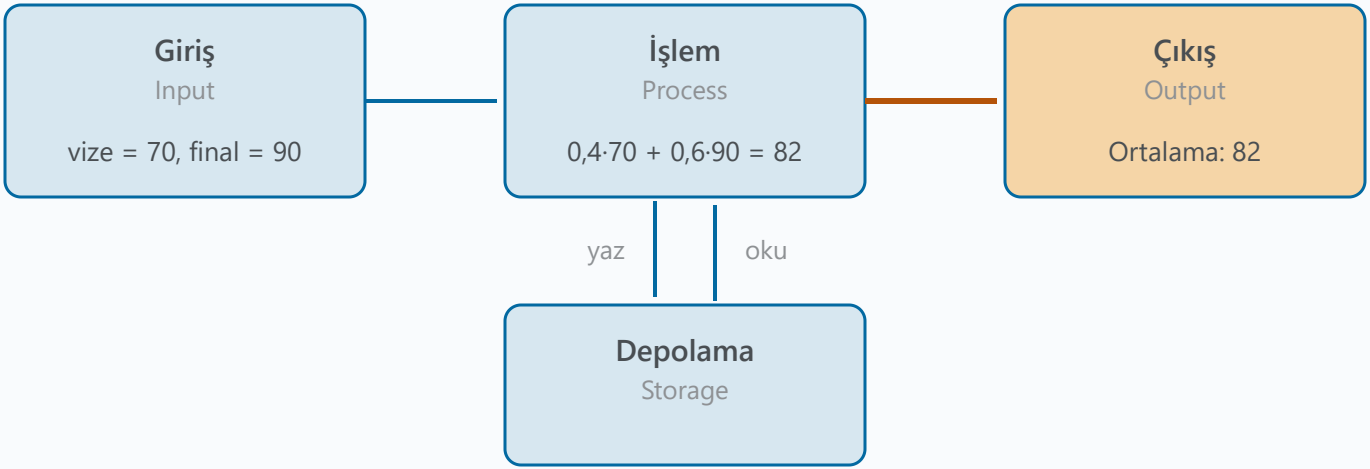
Giriş: Öğretmen formda vize = 70, final = 90 yazar. Klavye bu değerleri bilgisayara iletir.

Temel Çalışma Döngüsü



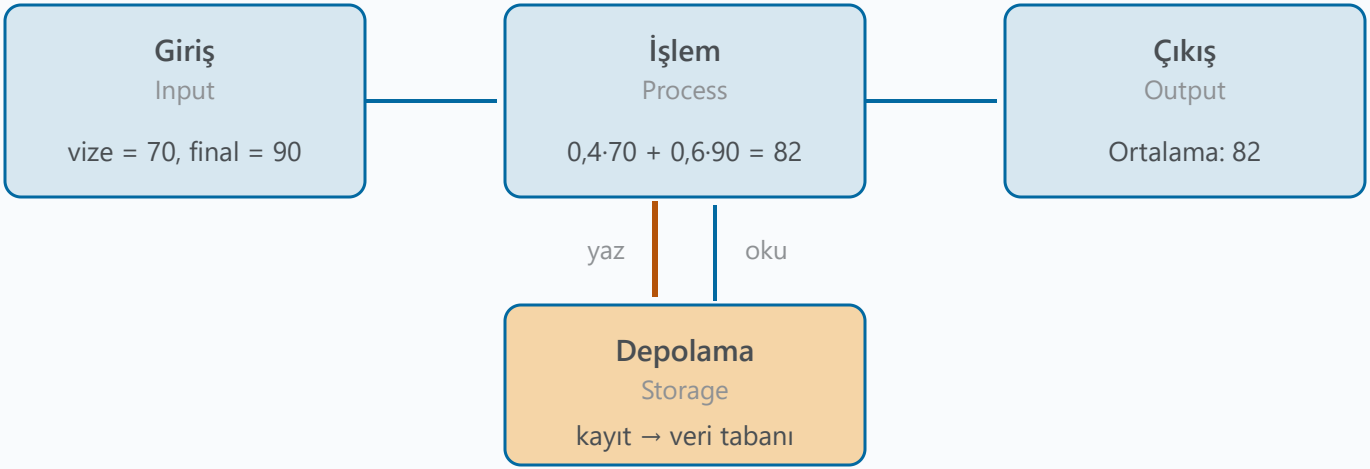
İşlem: CPU formülü uygular: 0,4 × 70 + 0,6 × 90 = **82**. Bu bir aritmetik işlemdir.

Temel Çalışma Döngüsü



Çıkış: Sonuç ekranda "Ortalama: 82" olarak görünür. Monitör çıkış birimidir.

Temel Çalışma Döngüsü



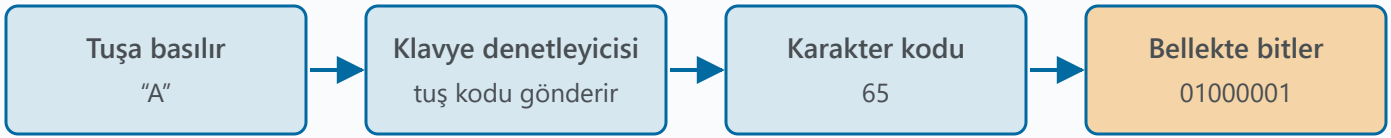
Depolama: Not kaydı veri tabanına yazılır. Daha sonra okunup yeniden işlenebilir (çift yönlü ok).

ÇALIŞMA PRENSİBİ

Giriş (Input)

Giriş, bilgisayarın kullanıcıdan, sensörden veya başka bir sistemden veri almasıdır.



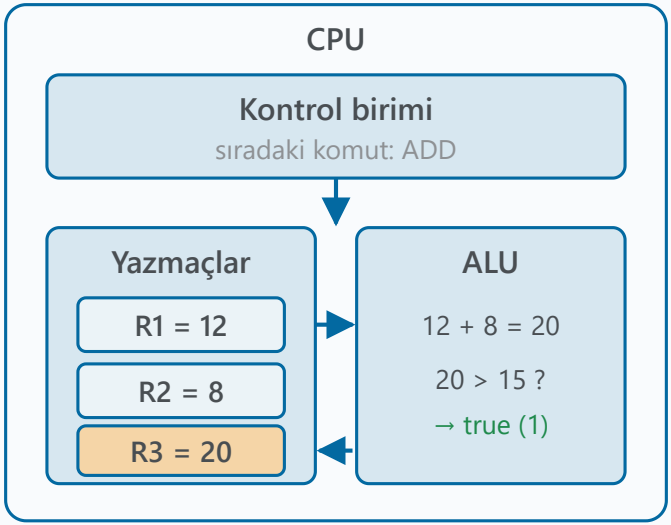


Her giriş birimi, fiziksel olayı sayılara (bitlere) çevirir.

ÇALIŞMA PRENSİBİ

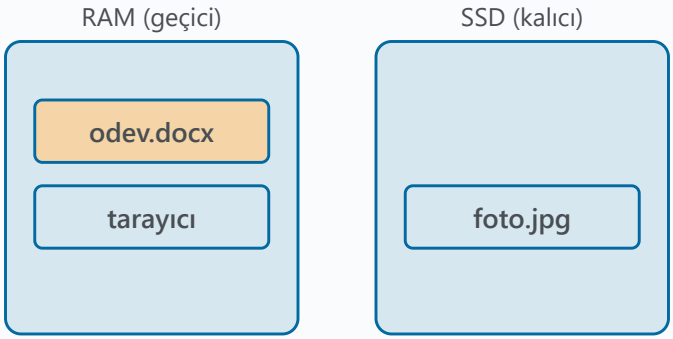
İşlem (Process)

- Merkezî işlem birimi (CPU), program komutlarını sırayla yürütür; **kontrol birimi** bu sırayı yönetir.
- **ALU** (aritmetik-mantık birimi) hesaplamaları yapar: 12 + 8 = 20.
- Mantıksal işlemler karşılaştırma ve karar vermeyi sağlar.
- Veriler ve komutlar ikili biçimde temsil edilir.



ÇALIŞMA PRENSİBİ

Çıkış ve Depolama



Güç: açık

Açık belge ve çalışan programlar **RAM**'de tutulur: hızlı erişim.

Adım 1 / 3

Çıkış

İşlenmiş bilgi monitör, yazıcı veya hoparlörle sunulur.

`hesap sonucu → ekran`

Geçici depolama (RAM)

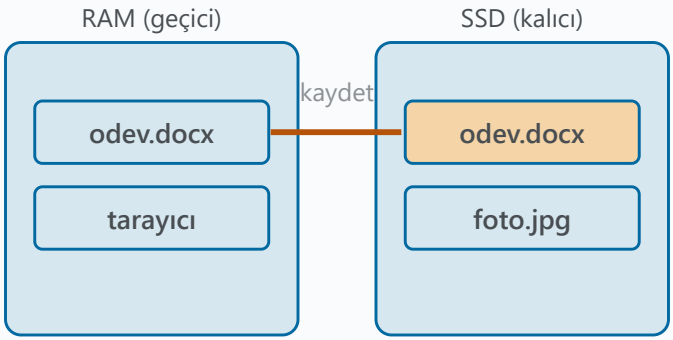
Hızlıdır; güç kesilince içeriği kaybolur.

Kalıcı depolama

SSD, sabit disk, USB bellek; veriyi güç olmadan da korur.

ÇALIŞMA PRENSİBİ

Çıkış ve Depolama



Güç: açık

Kaydet komutu belgeyi RAM'den **SSD**'ye kopyalar.

Adım 2 / 3

Çıkış

İşlenmiş bilgi monitör, yazıcı veya hoparlörle sunulur.

```
hesap sonucu → ekran
```

Geçici depolama (RAM)

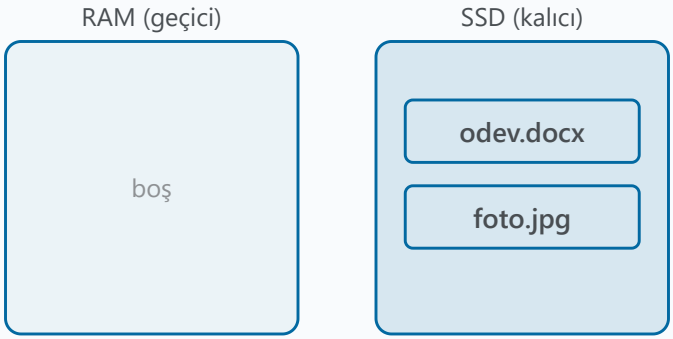
Hızlıdır; güç kesilince içeriği kaybolur.

Kalıcı depolama

SSD, sabit disk, USB bellek; veriyi güç olmadan da korur.

ÇALIŞMA PRENSİBİ

Çıkış ve Depolama



Güç kesilince RAM **silinir**; SSD'deki dosyalar **korunur**.

Adım 3 / 3

Çıkış

İşlenmiş bilgi monitör, yazıcı veya hoparlörle sunulur.

`hesap sonucu → ekran`

Geçici depolama (RAM)

Hızlıdır; güç kesilince içeriği kaybolur.

Kalıcı depolama

SSD, sabit disk, USB bellek; veriyi güç olmadan da korur.

SAYISAL TEMSİL

Her Veri Sayıya Dönüşür

Metin

Karakterler sayısal kodlarla temsil edilir.

`A → 65 → 01000001`

Fare hareketi

Ekrandaki konum koordinatlara çevrilir.

`(x, y) → (420, 180)`

Görüntü

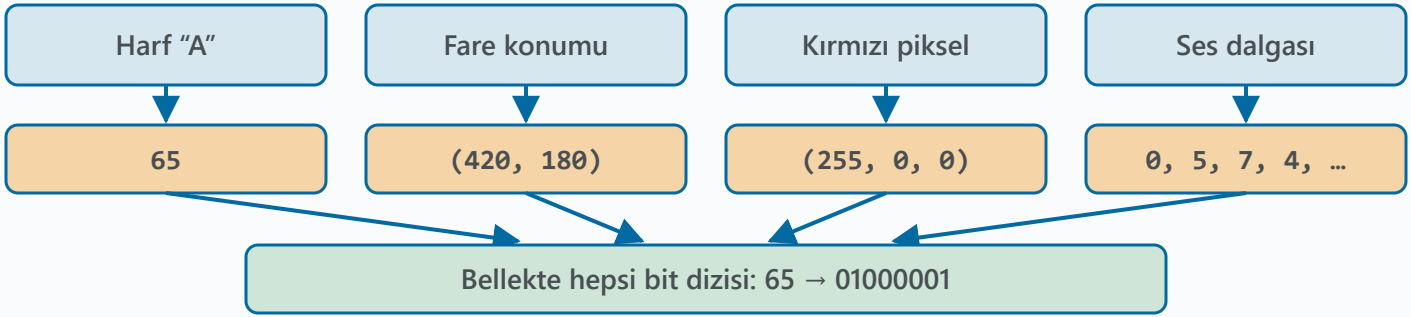
Her piksel renk bileşenlerinden oluşur.

`RGB(255, 0, 0)`

Ses

Ses dalgası belirli aralıklarla örneklenir.

`genlik → sayı dizisi`

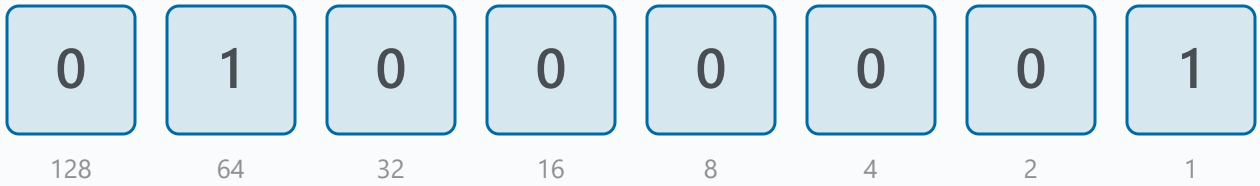


SAYISAL TEMSİL · ADIM ADIM

Bit ve İkili Sistem: İkiliden Onluğa

Bit, binary digit ifadesinin kısaltmasıdır ve yalnızca 0 ya da 1 değerini alır. 8 bit = 1 bayt.

Basamak değerleri (2'nin kuvvetleri):



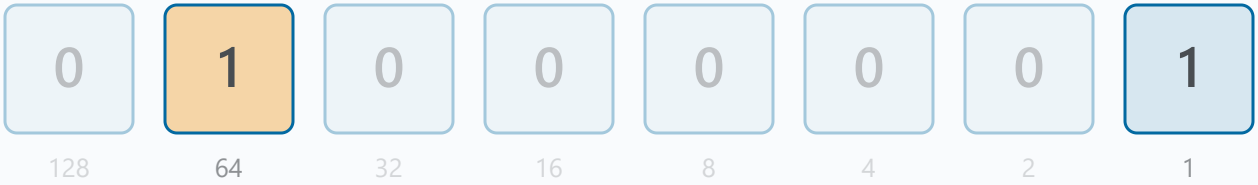
8 bitlik dizi: her konumun değeri, sağdan sola 1, 2, 4, … 128 diye **ikiye katlanır**.

Adım 1 / 4

Bit ve İkili Sistem: İkiliden Onluğa

Bit, binary digit ifadesinin kısaltmasıdır ve yalnızca 0 ya da 1 değerini alır. 8 bit = 1 bayt.

Basamak değerleri (2'nin kuvvetleri):



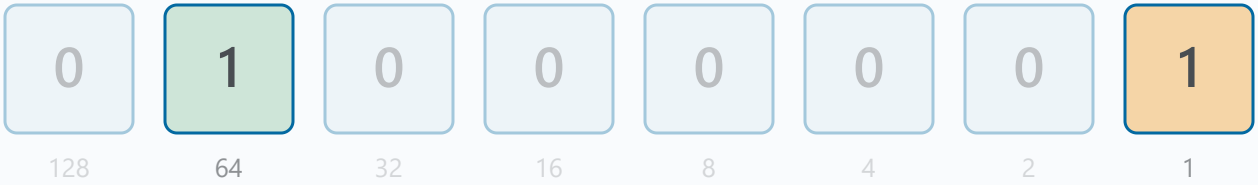
toplam = 64 = 64

6. konumdaki bit **1** → değeri **64** toplama eklenir. 0 olan bitler katkı yapmaz.

Bit ve İkili Sistem: İkiliden Onluğa

Bit, binary digit ifadesinin kısaltmasıdır ve yalnızca 0 ya da 1 değerini alır. 8 bit = 1 bayt.

Basamak değerleri (2'nin kuvvetleri):



toplam = 64 + 1 = 65

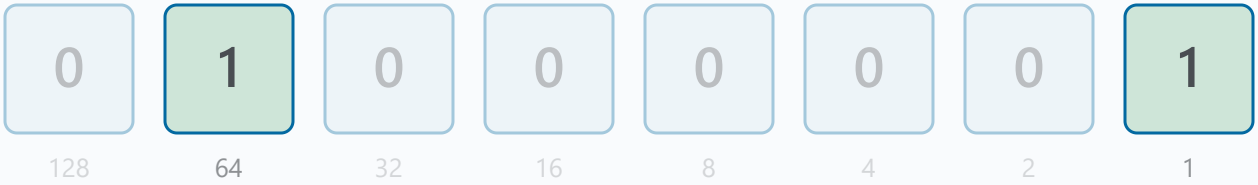
0. konumdaki bit **1** → değeri **1** toplama eklenir. 0 olan bitler katkı yapmaz.

SAYISAL TEMSİL · ADIM ADIM

Bit ve İkili Sistem: İkiliden Onluğa

Bit, binary digit ifadesinin kısaltmasıdır ve yalnızca 0 ya da 1 değerini alır. 8 bit = 1 bayt.

Basamak değerleri (2'nin kuvvetleri):



toplam = 64 + 1 = 65

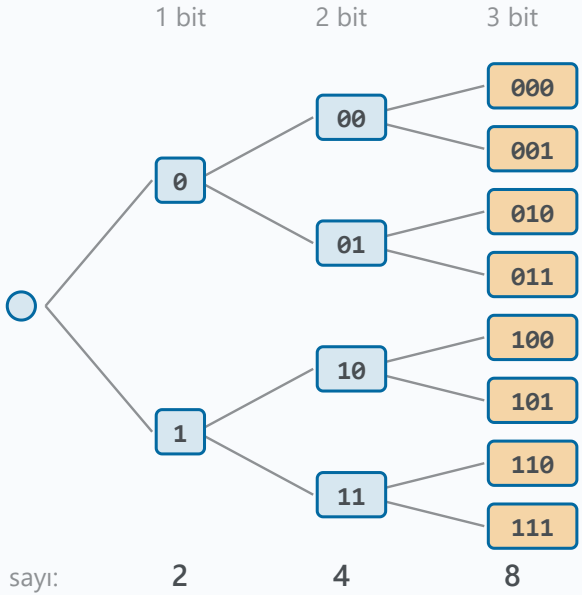
ASCII 65 → “A”

$01000001_2 = 65_{10}$. ASCII tablosunda 65 sayısı **A** harfine karşılık gelir.

İkili Basamakların Değeri

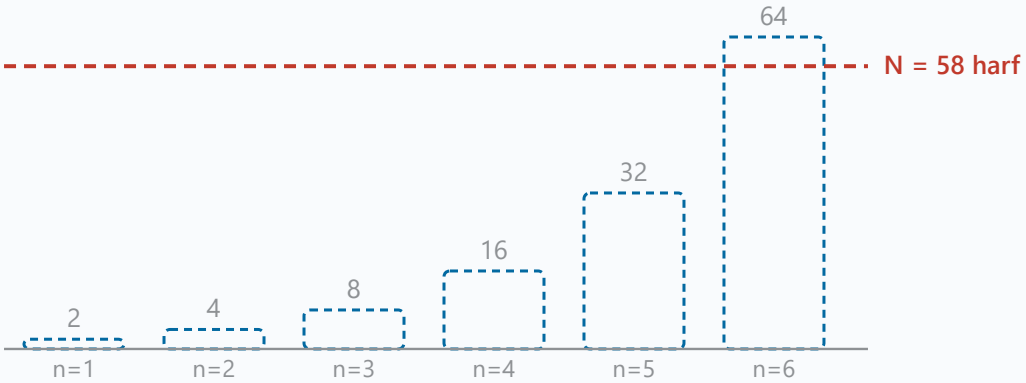
Bit sayısı	Kombinasyon	İşaretsiz değer aralığı
1 bit	$2^1 = 2$	0–1
2 bit	$2^2 = 4$	0–3
8 bit	$2^8 = 256$	0–255
10 bit	$2^{10} = 1.024$	0–1.023
32 bit	2^{32}	0–4.294.967.295

Her yeni bit, temsil edilebilen farklı değer sayısını iki katına çıkarır: n bit → 2^n değer.



SAYISAL TEMSİL · ADIM ADIM

Kaç Bit Gerekir? 58 Harf



Türk alfabesinde 29 harf var; büyük ve küçük harfler ayrı kodlanırsa **N = 58** farklı değer gerekir. Soru: en az kaç bit?

Adım 1 / 8

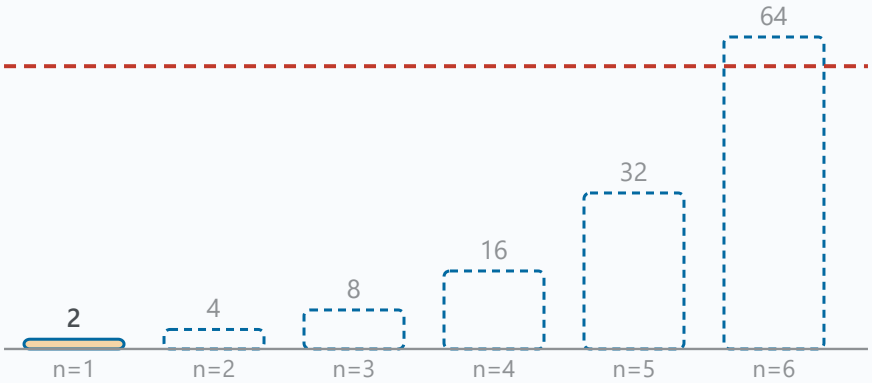
```
▶ N = 58   (29 büyük + 29 küçük harf)
2 n = 1; kapasite = 2
3 kapasite < N iken: n = n + 1; kapasite =
  kapasite × 2
4 kapasite ≥ N → n bit yeter
```

DEĞİŞKENLER

`N = 58` `n = –` `kapasite = –`

SAYISAL TEMSİL · ADIM ADIM

Kaç Bit Gerekir? 58 Harf



1 bit → 2 kod

2 < 58 → yetmez

Tek bit yalnızca 2 kod verir (0 ve 1). 2 < 58 olduğu için döngü sürer.

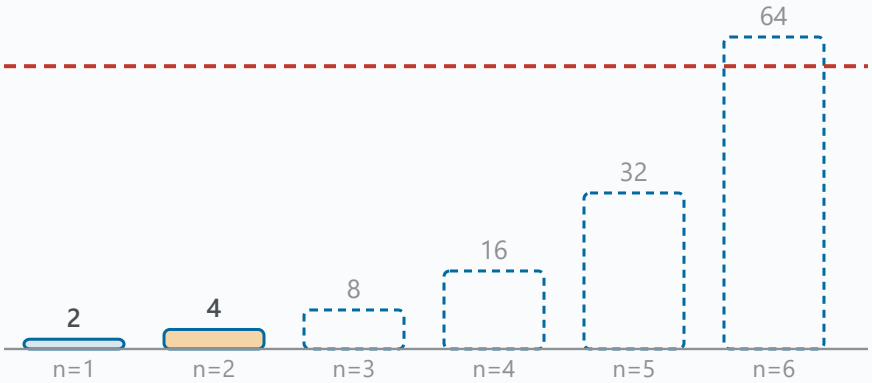
Adım 2 / 8

```
1 N = 58   (29 büyük + 29 küçük harf)
▶ n = 1; kapasite = 2
3 kapasite < N iken: n = n + 1; kapasite =
  kapasite × 2
4 kapasite ≥ N → n bit yeter
```

DEĞİŞKENLER

`N = 58` `n = 1` `kapasite = 2`

Kaç Bit Gerekir? 58 Harf



2 bit → 4 kod

4 < 58 → yetmez

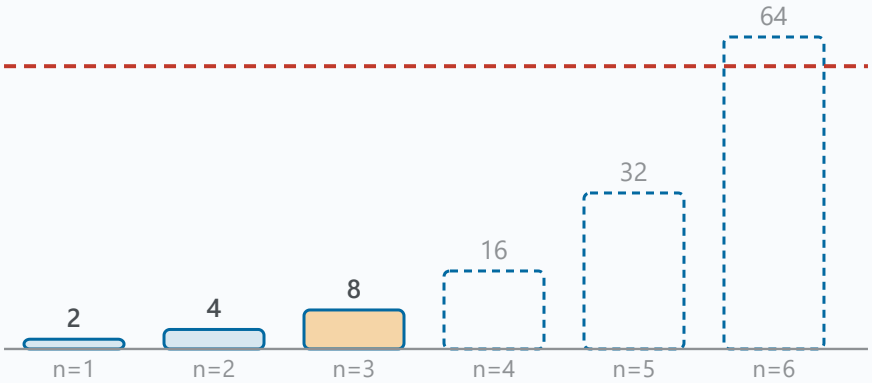
Bir bit eklenince kod sayısı ikiye katlanır: **4**. Hâlâ 58'den az.


```
1 N = 58   (29 büyük + 29 küçük harf)
2 n = 1; kapasite = 2
▶ kapasite < N iken: n = n + 1; kapasite =
  kapasite × 2
4 kapasite ≥ N → n bit yeter
```

DEĞİŞKENLER

`N = 58` `n = 2` `kapasite = 4`

Kaç Bit Gerekir? 58 Harf



3 bit → 8 kod

8 < 58 → yetmez

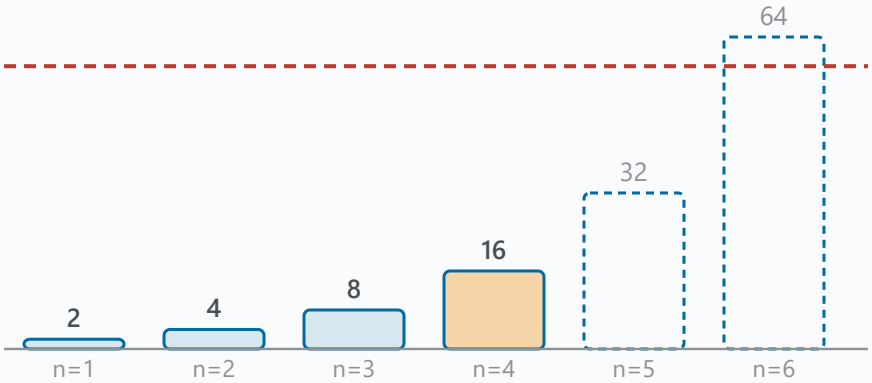
Bir bit eklenince kod sayısı ikiye katlanır: **8**. Hâlâ 58'den az.


```
1 N = 58   (29 büyük + 29 küçük harf)
2 n = 1; kapasite = 2
▶ kapasite < N iken: n = n + 1; kapasite =
  kapasite × 2
4 kapasite ≥ N → n bit yeter
```

DEĞİŞKENLER

`N = 58` `n = 3` `kapasite = 8`

Kaç Bit Gerekir? 58 Harf



4 bit → 16 kod

16 < 58 → yetmez

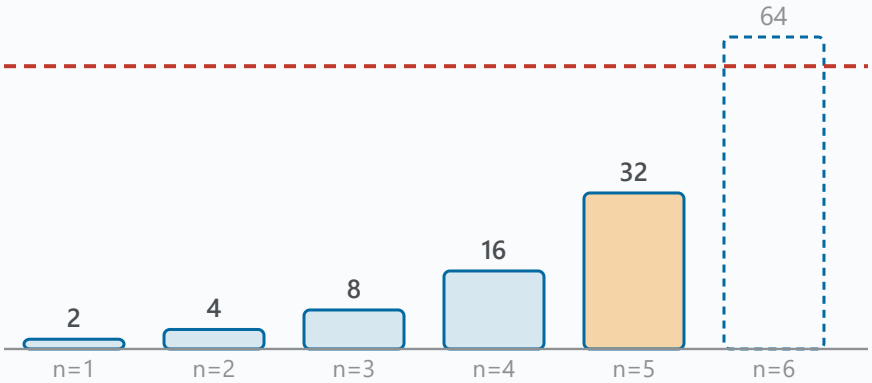
Bir bit eklenince kod sayısı ikiye katlanır: **16**. Hâlâ 58'den az.


```
1 N = 58   (29 büyük + 29 küçük harf)
2 n = 1; kapasite = 2
▶ kapasite < N iken: n = n + 1; kapasite =
  kapasite × 2
4 kapasite ≥ N → n bit yeter
```

DEĞİŞKENLER

`N = 58` `n = 4` `kapasite = 16`

Kaç Bit Gerekir? 58 Harf



5 bit → 32 kod

32 < 58 → yetmez

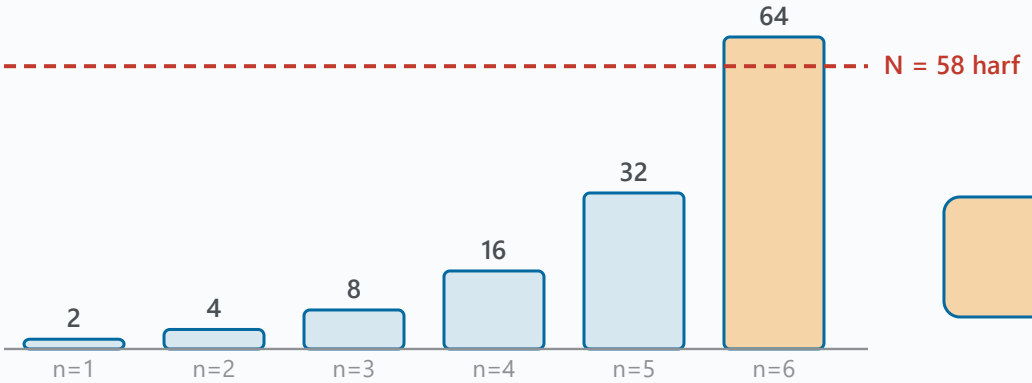
Bir bit eklenince kod sayısı ikiye katlanır: **32**. Hâlâ 58'den az.


```
1 N = 58   (29 büyük + 29 küçük harf)
2 n = 1; kapasite = 2
▶ kapasite < N iken: n = n + 1; kapasite =
  kapasite × 2
4 kapasite ≥ N → n bit yeter
```

DEĞİŞKENLER

`N = 58` `n = 5` `kapasite = 32`

Kaç Bit Gerekir? 58 Harf



6 bit → 64 kod
64 ≥ 58 → yeter

6 bit → **64** kod. 64 ≥ 58: koşul artık sağlanmıyor, döngü durur.

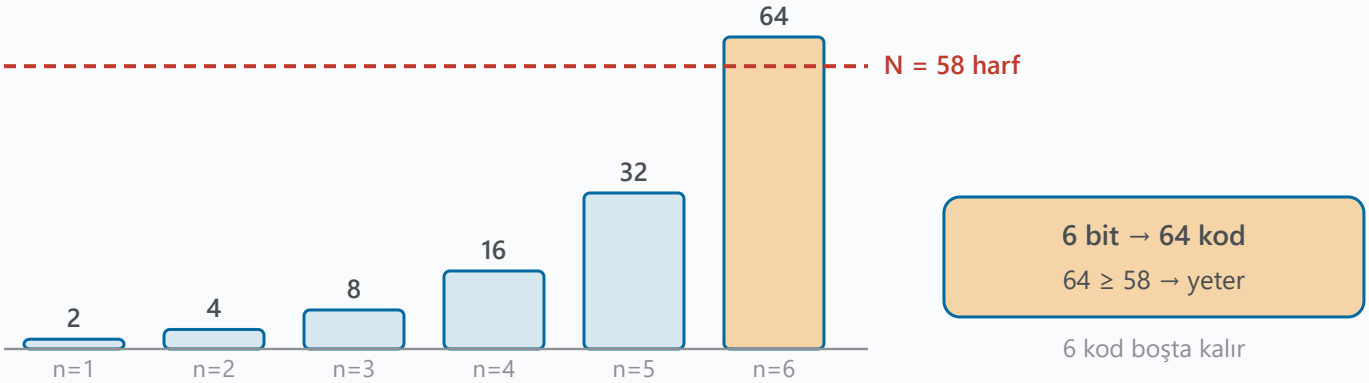
Adım 7 / 8

```
1 N = 58   (29 büyük + 29 küçük harf)
2 n = 1; kapasite = 2
▶ kapasite < N iken: n = n + 1; kapasite =
  kapasite × 2
4 kapasite ≥ N → n bit yeter
```

DEĞİŞKENLER

`N = 58` `n = 6` `kapasite = 64`

Kaç Bit Gerekir? 58 Harf



Sonuç **6 bit**; 64 − 58 = 6 kod kullanılmaz. Rakamlar ve noktalama da eklenince 64 yetmez, 7 bit (128) gerekir: ASCII bu yüzden 7 bitliktir.

Adım 8 / 8

```
1 N = 58   (29 büyük + 29 küçük harf)
2 n = 1; kapasite = 2
3 kapasite < N iken: n = n + 1; kapasite =
  kapasite × 2
▶ kapasite ≥ N → n bit yeter
```

DEĞİŞKENLER

`N = 58` `n = 6` `kapasite = 64` `boşta = 6`

SAYISAL TEMSİL · ADIM ADIM

Onluktan İkiliye Dönüşüm: 45

Sürekli 2'ye böl, kalanı yaz

Kalanlar alttan üste okunur



45 sayısını ikiliye çevireceğiz. Yöntem: bölüm 0 olana kadar 2'ye böl, her adımda kalanı (0 ya da 1) not al.

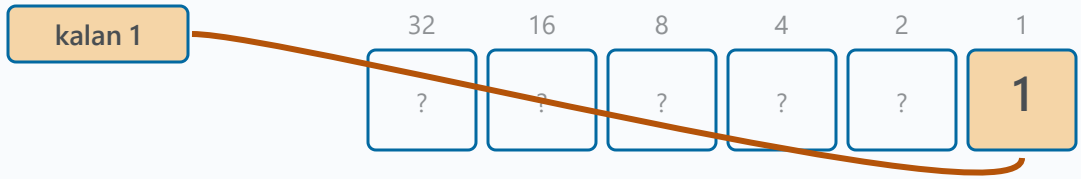
Adım 1 / 8

Onluktan İkiliye Dönüşüm: 45

Sürekli 2'ye böl, kalanı yaz

Kalanlar alttan üste okunur

45 ÷ 2 = 22



45 ÷ 2 = 22, **kalan 1** (tek sayı → kalan 1). İlk kalan en sağdaki (1'ler) basamaktır.

Onluktan İkiliye Dönüşüm: 45

Sürekli 2'ye böl, kalanı yaz

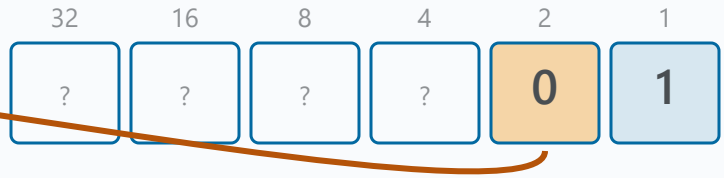
45 ÷ 2 = 22

22 ÷ 2 = 11

kalan 1

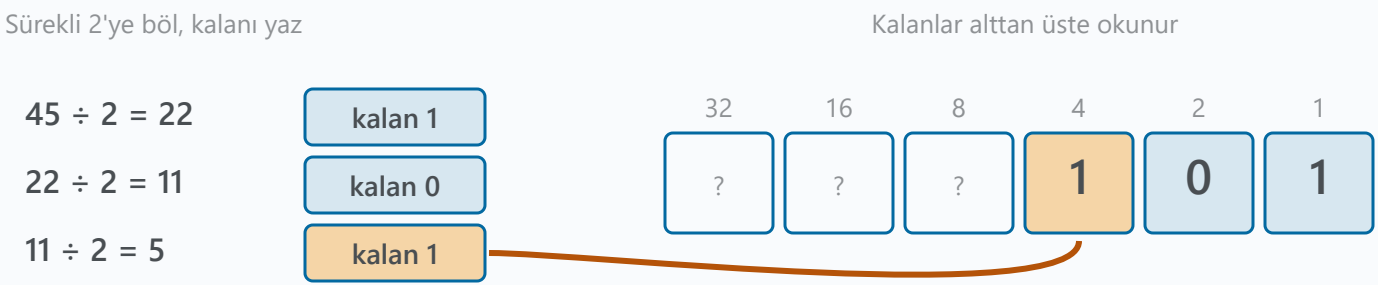
kalan 0

Kalanlar alttan üste okunur



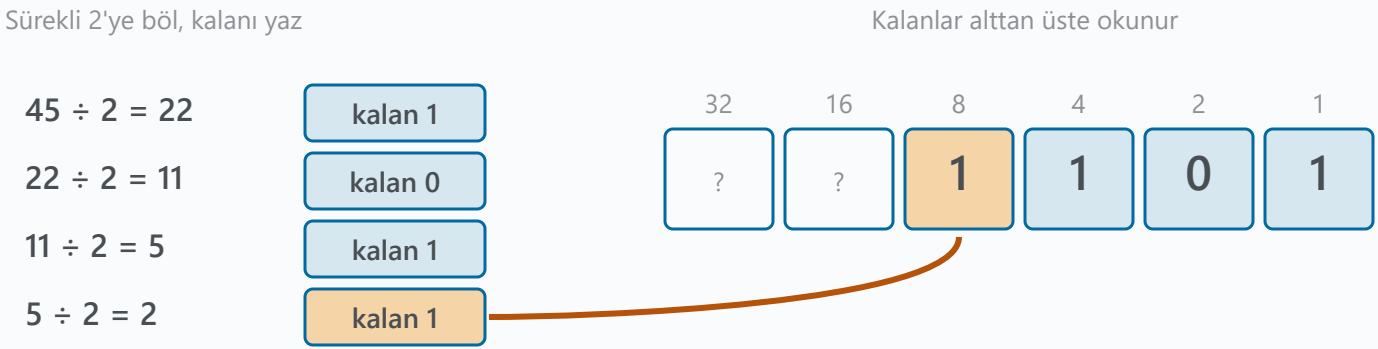
22 ÷ 2 = 11, **kalan 0** (çift sayı → kalan 0). Bu kalan bir soldaki basamağa yazılır.

Onluktan İkiliye Dönüşüm: 45



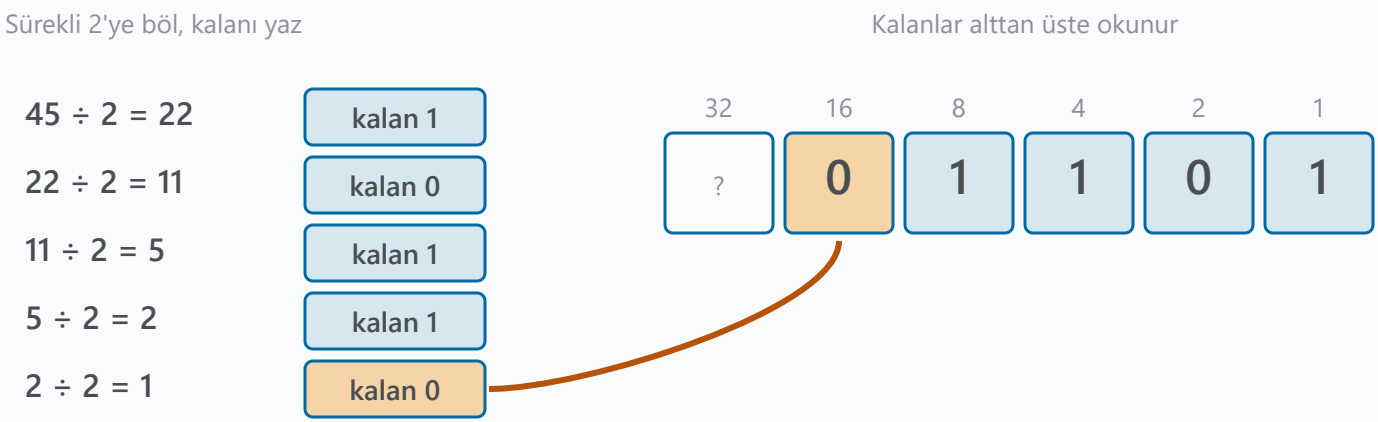
11 ÷ 2 = 5, **kalan 1** (tek sayı → kalan 1). Bu kalan bir soldaki basamağa yazılır.

Onluktan İkiliye Dönüşüm: 45



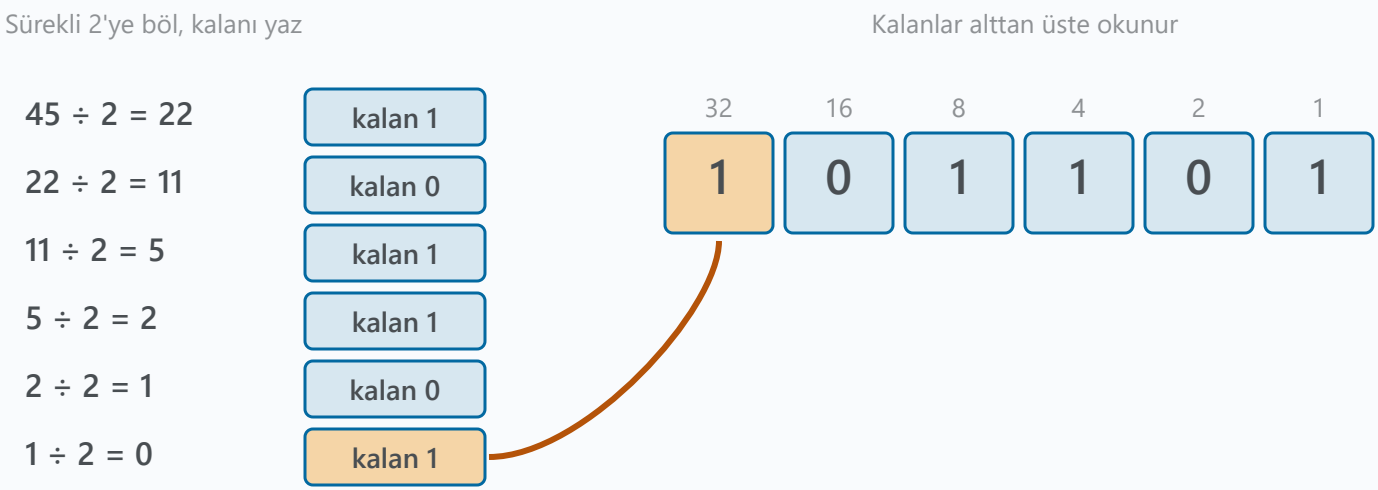
5 ÷ 2 = 2, **kalan 1** (tek sayı → kalan 1). Bu kalan bir soldaki basamağa yazılır.

Onluktan İkiliye Dönüşüm: 45



2 ÷ 2 = 1, **kalan 0** (çift sayı → kalan 0). Bu kalan bir soldaki basamağa yazılır.

Onluktan İkiliye Dönüşüm: 45



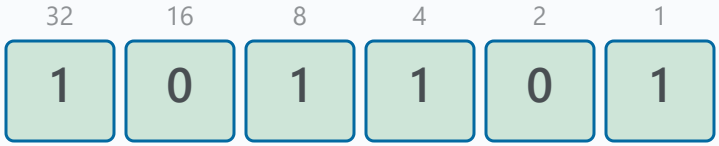
1 ÷ 2 = 0, **kalan 1** (tek sayı → kalan 1). Bu kalan bir soldaki basamağa yazılır. Bölüm 0 oldu, bölme biter.

Onluktan İkiliye Dönüşüm: 45

Sürekli 2'ye böl, kalanı yaz

45 ÷ 2 = 22	kalan 1
22 ÷ 2 = 11	kalan 0
11 ÷ 2 = 5	kalan 1
5 ÷ 2 = 2	kalan 1
2 ÷ 2 = 1	kalan 0
1 ÷ 2 = 0	kalan 1

Kalanlar alttan üste okunur



101101_2

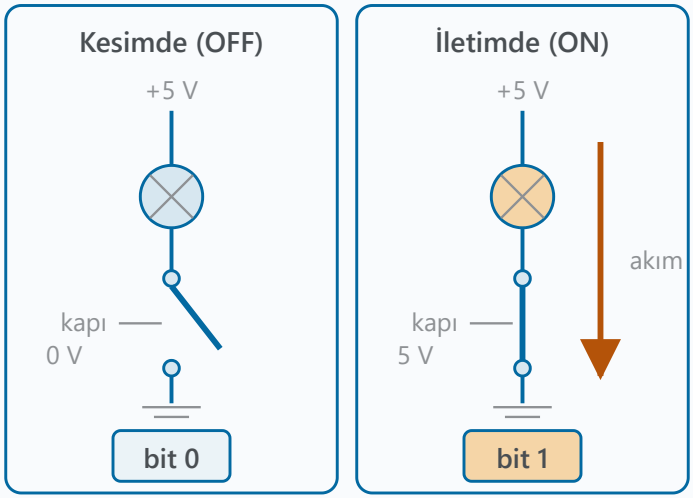
kontrol: 32 + 8 + 4 + 1 = 45

Kalanlar **son bölmeden ilke** okununca **$45_{10} = 101101_2$**. Doğrulama: basamak değerlerini topla.

SAYISAL TEMSİL

Transistörler

- İşlemciler çok sayıda transistörden oluşur.
- Transistör, elektrik akımını kontrol eden çok küçük bir anahtar gibi çalışır.
- Kapı gerilimi transistörü **iletime** (akım geçer) veya **kesime** (akım geçmez) sokar; bu iki durum 1 ve 0 olarak yorumlanır.
- Milyarlarca anahtarın düzenli çalışması karmaşık işlemleri mümkün kılar.



VERİ TEMSİLİ

Metin ve Sayılar

Sayılar

Tam ve kayan noktalı sayılar, belirli ikili kodlama kurallarıyla saklanır.

`13 → 00001101`

Karakterler

ASCII ve Unicode, karakterlerle sayısal kodları eşleştirir.

`A → U+0041`

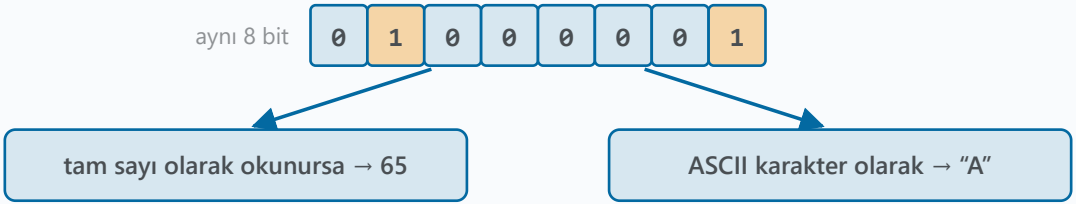
Kodlama

Aynı bit dizisinin anlamını kullanılan kodlama standardı belirler.

Unicode

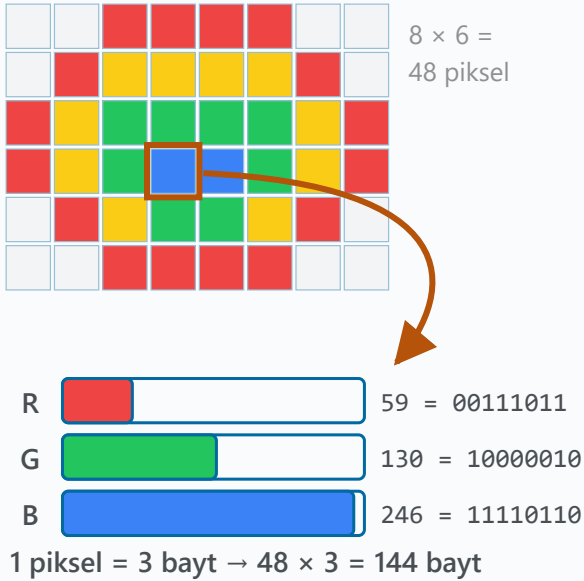
Farklı alfabeler ve simgeler için geniş bir karakter kümesi sunar.

`Ğ · 中 · €`



Görüntüler

- Dijital görüntü, piksel adı verilen küçük renk hücrelerinden oluşur.
- Çözünürlük, yatay ve dikey piksel sayısını belirtir.
- RGB modelinde renk; kırmızı, yeşil ve mavi bileşenlerle tanımlanır.
- Her bileşen 0–255 arası bir sayıdır (1 bayt). `RGB(255, 255, 255)` beyaz, `RGB(0, 0, 0)` siyahtır.



VERİ TEMSİLİ

Renk Derinliği ve Görüntü Boyutu

Bit / piksel	Farklı renk	Tipik kullanım
1	$2^1 = 2$	Siyah-beyaz çizim, faks, tarama
8	$2^8 = 256$	Gri tonlama ya da paletli görüntü (GIF)
24	$2^{24} = 16.777.216$	Fotoğraf: R, G, B için 8'er bit
32	16,7 milyon + saydamlık	Arayüz grafikleri, PNG (alfa kanalı)

Boyut formülü

Sıkıştırmasız görüntü = genişlik × yükseklik × bit derinliği ÷ 8 bayt.

Örnek: 800 × 600

24 bit: 800 × 600 × 3 = 1.440.000 bayt ≈ 1,37 MB. Aynı görüntü 1 bitle: 60.000 bayt ≈ 58,6 KB.

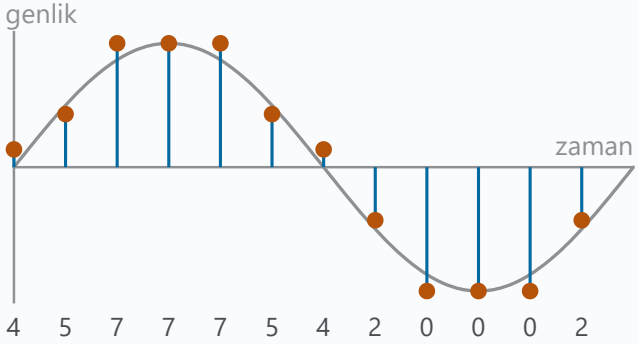
Renk derinliği arttıkça renk geçişleri yumuşar ama dosya büyür; 8 bitlik palette gökyüzü gibi geçişler basamaklı görünür.

VERİ TEMSİLİ

Ses ve Video

Ses

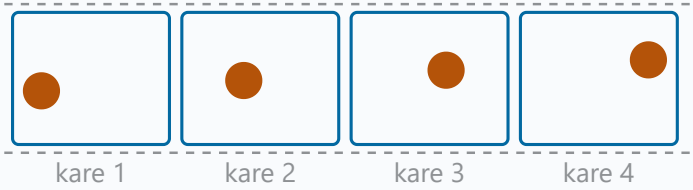
Analog dalga eşit aralıklarla ölçülür (**örnekleme**); her ölçüm bir sayıya yuvarlanır.



Örnekler: 4, 5, 7, 7, 7, 5, 4, 2, 0, 0, 0, 2

Video

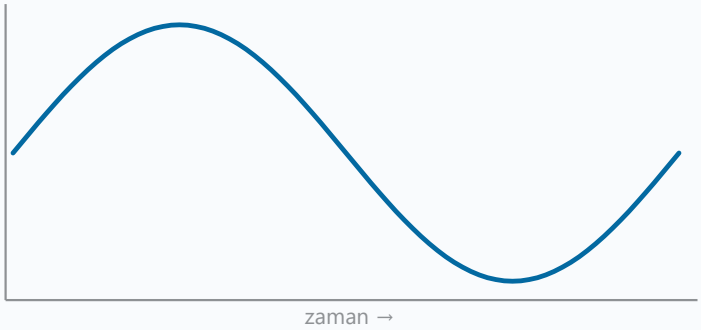
Görüntü karelerinin art arda gösterilmesi hareket algısı oluşturur. Ses çoğunlukla ayrı bir kanalda saklanır.



30 fps: saniyede 30 kare

VERİ TEMSİLİ · ADIM ADIM

Sesi Sayısallaştırma: Örnekle, Nicemle, Kodla



Analog sinyal

mikrofon gerilimi

zamanla sürekli değişir

her an bir değer var:

sonsuz sayıda nokta

bilgisayar bunu saklayamaz

Mikrofon sesi sürekli değişen bir gerilime çevirir. Bu **analog** sinyalin her anında bir değer vardır; sonsuz sayıda değer saklanamaz.

Adım 1 / 6

```
▶ analog sinyal: sürekli dalga
2 örnekle: eşit aralıklarla ölç
3 nicemle: en yakın düzeye yuvarla (3 bit → 0…7)
4 kodla: her düzeyi 3 bitle yaz
5 boyut = örnekleme hızı × bit derinliği × kanal ×
  süre
6 kayıplı sıkıştır (MP3)
```

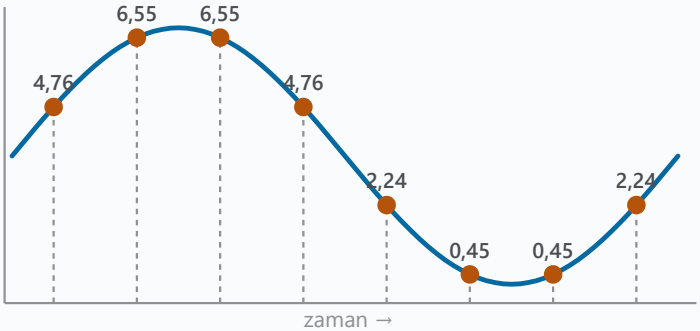
DEĞİŞKENLER

örnek sayısı = –

düzeyler = –

bit = –

Sesi Sayısallaştırma: Örnekle, Nicemle, Kodla



Örnekleme

eşit aralıklarla 8 ölçüm

ölçümler arası kaybolur

örnekleme hızı: saniyedeki ölçüm sayısı (Hz)

CD: 44.100 Hz

Örnekleme: sinyal eşit aralıklarla ölçülür. Ölçümler ondalıklıdır (4,76; 6,55…); iki ölçüm arasındaki değişim kaydedilmez.

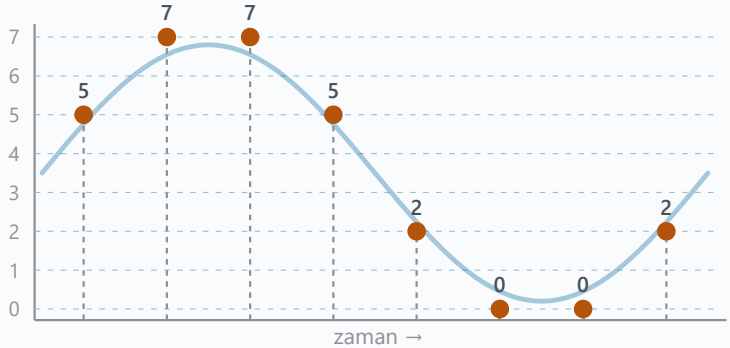
Adım 2 / 6

```
1 analog sinyal: sürekli dalga
▶ örnekle: eşit aralıklarla ölç
3 nicemle: en yakın düzeye yuvarla (3 bit → 0…7)
4 kodla: her düzeyi 3 bitle yaz
5 boyut = örnekleme hızı × bit derinliği × kanal ×
  süre
6 kayıplı sıkıştır (MP3)
```

DEĞİŞKENLER

örnek sayısı = 8 | düzeyler = – | bit = –

Sesi Sayısallaştırma: Örnekle, Nicemle, Kodla



Nicemleme

3 bit → 2^3 = 8 düzey (0…7)

her örnek en yakın düzeye yuvarlanır: 4,76 → 5

en büyük hata: 0,45 düzey

16 bit → 65.536 düzey

Nicemleme: 3 bit yalnızca 8 düzey tanır; her ölçüm en yakın düzeye yuvarlanır. Aradaki fark (en çok 0,45) nicemleme hatasıdır.

Adım 3 / 6

```
1 analog sinyal: sürekli dalga
2 örnekle: eşit aralıklarla ölç
▶ nicemle: en yakın düzeye yuvarla (3 bit → 0…7)
4 kodla: her düzeyi 3 bitle yaz
5 boyut = örnekleme hızı × bit derinliği × kanal × süre
6 kayıplı sıkıştır (MP3)
```

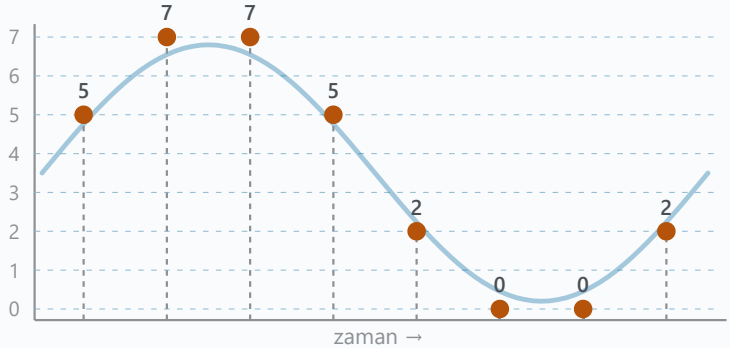
DEĞİŞKENLER

örnek sayısı = **8**

düzeyler = **5,7,7,5,2,0,0,2**

bit = **–**

Sesi Sayısallaştırma: Örnekle, Nicemle, Kodla



3 bitlik kodlar

101	111	111	101
010	000	000	010

8 örnek × 3 bit = 24 bit

Kodlama: her düzey 3 bitlik ikili sayıyla yazılır; 5 → 101, 7 → 111. Sekiz örnek 24 bit tutar.

Adım 4 / 6

```
1 analog sinyal: sürekli dalga
2 örnekle: eşit aralıklarla ölç
3 nicemle: en yakın düzeye yuvarla (3 bit → 0…7)
▶ kodla: her düzeyi 3 bitle yaz
5 boyut = örnekleme hızı × bit derinliği × kanal × süre
6 kayıplı sıkıştır (MP3)
```

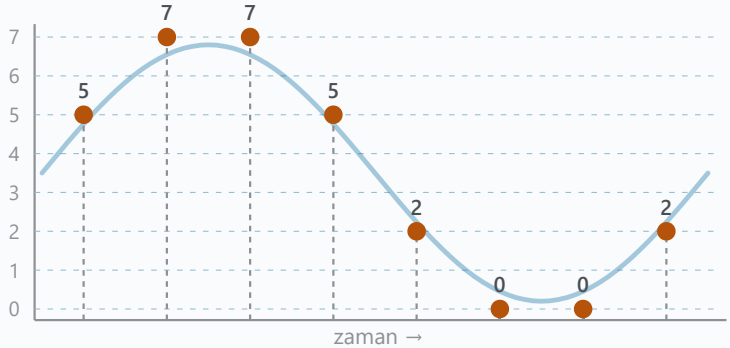
DEĞİŞKENLER

örnek sayısı = **8**

düzeyler = **5,7,7,5,2,0,0,2**

bit = **24**

Sesi Sayısallaştırma: Örnekle, Nicemle, Kodla



CD kalitesi, 1 dakika

44.100 örnek/s × 16 bit × 2 kanal

= 1.411.200 bit/s

× 60 s = 84.672.000 bit

÷ 8 = 10.584.000 bayt

≈ **10,1 MB**

Gerçek ses saniyede binlerce örnek alır. CD kalitesi: 44.100 Hz, 16 bit, 2 kanal → bir dakika ≈ **10,1 MB** (1 MB = 1.048.576 bayt).

Adım 5 / 6

```
1 analog sinyal: sürekli dalga
2 örnekle: eşit aralıklarla ölç
3 nicemle: en yakın düzeye yuvarla (3 bit → 0…7)
4 kodla: her düzeyi 3 bitle yaz
▶ boyut = örnekleme hızı × bit derinliği × kanal × süre
6 kayıplı sıkıştır (MP3)
```

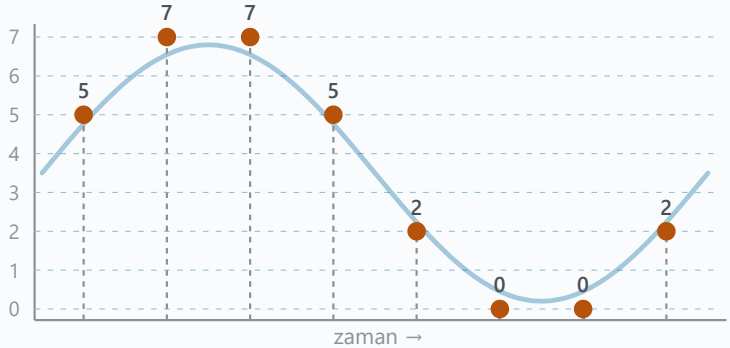
DEĞİŞKENLER

`örnekleme hızı = 44.100 Hz`

`bit derinliği = 16` `kanal = 2`

`1 dk = 10.584.000 bayt`

Sesi Sayısallaştırma: Örnekle, Nicemle, Kodla



1 dakika müzik

Sıkıştırmasız (WAV)

≈ 10,1 MB

MP3, 128 kbit/s

960.000 bayt ≈ 0,92 MB

MP3 gibi **kayıplı** biçimler kulağın zor fark ettiği ayrıntıları atar: 128 kbit/s ile bir dakika 960.000 bayt, yaklaşık 11 kat küçük.

Adım 6 / 6

```
1 analog sinyal: sürekli dalga
2 örnekle: eşit aralıklarla ölç
3 nicemle: en yakın düzeye yuvarla (3 bit → 0…7)
4 kodla: her düzeyi 3 bitle yaz
5 boyut = örnekleme hızı × bit derinliği × kanal ×
  süre
▶ kayıplı sıkıştır (MP3)
```

DEĞİŞKENLER

`WAV 1 dk = 10.584.000 bayt`

`MP3 1 dk = 960.000 bayt`

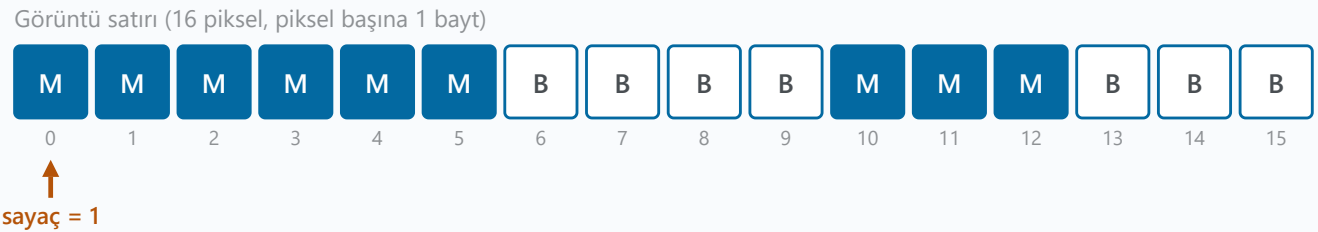
VERİ TEMSİLİ

Kayıpsız ve Kayıplı Sıkıştırma

	Kayıpsız	Kayıplı
Açılınca	Özgün veri bit bit aynen geri gelir	Özgüne yakın bir kopya elde edilir
Yöntem	Tekrarları kısaltır (RLE, ZIP)	Göz ve kulağın zor fark ettiği ayrıntıyı atar
Örnek biçimler	ZIP, PNG, FLAC	JPEG, MP3, MP4 (H.264)
Tipik küçülme	Çoğunlukla 2–3 kat	10 kat ve daha fazlası
Uygun veri	Metin, program, tablo, tıbbi görüntü	Fotoğraf, müzik, video

Kayıplı bir dosya her düzenleyip yeniden kaydedişte biraz daha bozulur; atılan ayrıntı geri gelmez. Bir programın ya da metnin tek biti bile değişse dosya işe yaramaz, bu yüzden onlar yalnızca kayıpsız sıkıştırılır.

Çalışma Uzunluğu Kodlaması (RLE)



Satırda iki renk var: **M** (mavi) ve **B** (beyaz). İlk pikselde sayaç 1, geçerli renk M.

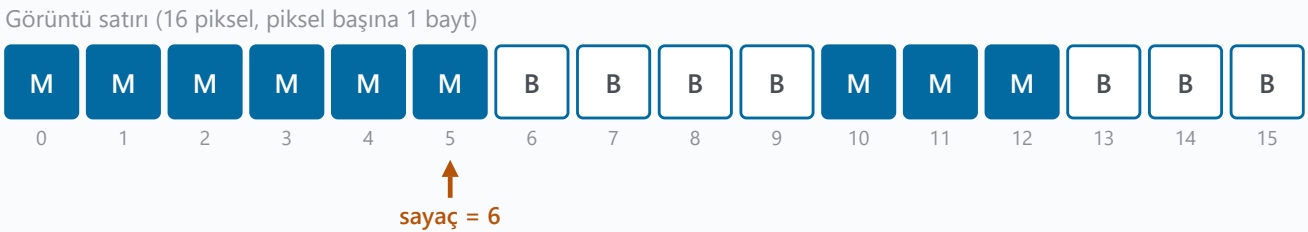
Adım 1 / 7

```
▶ sayaç = 1; renk = ilk piksel
2 sonraki piksel aynı renkse: sayaç = sayaç + 1
3 renk değişince: (sayaç, renk) yaz; sayaç = 1
4 satır bitince: son çifti yaz
5 boyut: 16 bayt → 4 çift × 2 bayt
6 kötü durum: M B M B …
```

DEĞİŞKENLER

`sayaç = 1` `renk = M` `çıktı = –`

Çalışma Uzunluğu Kodlaması (RLE)



Çıktı:

1–5 numaralı pikseller de M: her birinde sayaç bir artar. 5. pikselde sayaç = 6.

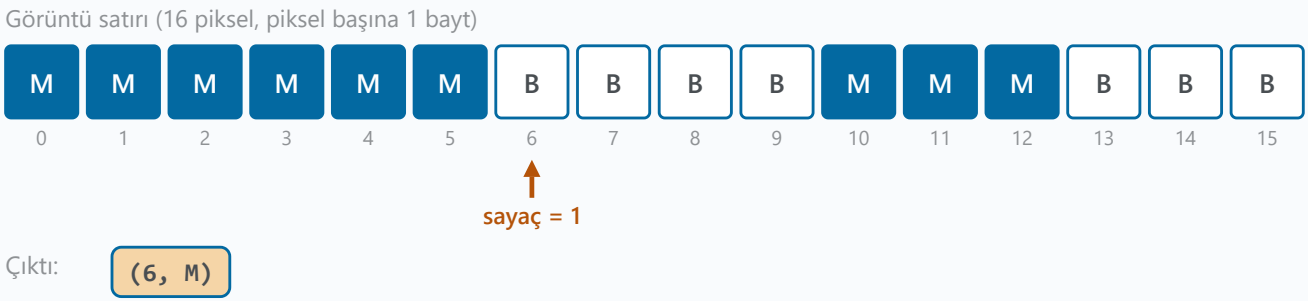
Adım 2 / 7

```
1 sayaç = 1; renk = ilk piksel
▶ sonraki piksel aynı renkse: sayaç = sayaç + 1
3 renk değişince: (sayaç, renk) yaz; sayaç = 1
4 satır bitince: son çifti yaz
5 boyut: 16 bayt → 4 çift × 2 bayt
6 kötü durum: M B M B …
```

DEĞİŞKENLER

`sayaç = 6` `renk = M` `çıktı = –`

Çalışma Uzunluğu Kodlaması (RLE)



6. piksel B: renk değişti. Biten koşu **(6, M)** olarak yazılır; sayaç 1'e döner, yeni renk B.

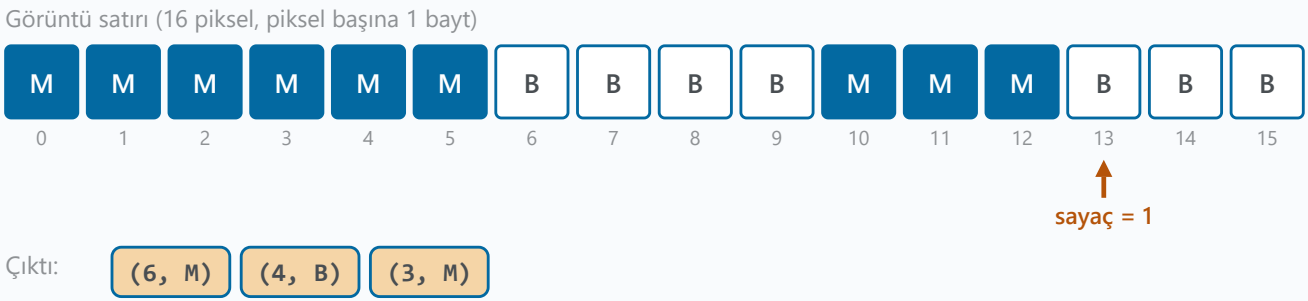
Adım 3 / 7

```
1 sayaç = 1; renk = ilk piksel
2 sonraki piksel aynı renkse: sayaç = sayaç + 1
▶ renk değişince: (sayaç, renk) yaz; sayaç = 1
4 satır bitince: son çifti yaz
5 boyut: 16 bayt → 4 çift × 2 bayt
6 kötü durum: M B M B …
```

DEĞİŞKENLER

`sayaç = 1` `renk = B` `çıktı = (6,M)`

Çalışma Uzunluğu Kodlaması (RLE)



Aynı kural iki kez daha işledi: dört B → (4, B), üç M → (3, M). 13. pikselde yeni bir B koşusu başladı.

Adım 4 / 7

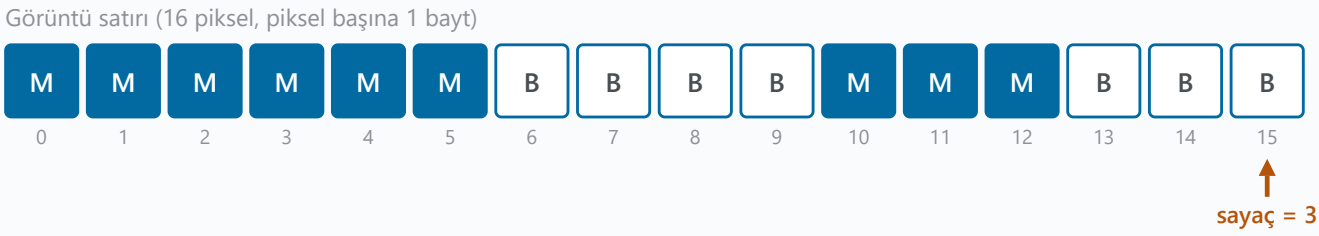
```
1 sayaç = 1; renk = ilk piksel
2 sonraki piksel aynı renkse: sayaç = sayaç + 1
▶ renk değişince: (sayaç, renk) yaz; sayaç = 1
4 satır bitince: son çifti yaz
5 boyut: 16 bayt → 4 çift × 2 bayt
6 kötü durum: M B M B …
```

DEĞİŞKENLER

`sayaç = 1` `renk = B`

`çıktı = (6,M) (4,B) (3,M)`

Çalışma Uzunluğu Kodlaması (RLE)



Satır bitti: açık kalan son koşu **(3, B)** de yazılır. Çıktıda 4 çift var.

Adım 5 / 7

```
1 sayaç = 1; renk = ilk piksel
2 sonraki piksel aynı renkse: sayaç = sayaç + 1
3 renk değişince: (sayaç, renk) yaz; sayaç = 1
▶ satır bitince: son çifti yaz
5 boyut: 16 bayt → 4 çift × 2 bayt
6 kötü durum: M B M B …
```

DEĞİŞKENLER

`sayaç = 3` `renk = B`

`çıktı = (6,M) (4,B) (3,M) (3,B)`

Çalışma Uzunluğu Kodlaması (RLE)

Görüntü satırı (16 piksel, piksel başına 1 bayt)



Çıktı: `(6, M)` `(4, B)` `(3, M)` `(3, B)`

Çözme: (6, M) → MMMMMM, (4, B) → BBBB … satır birebir geri gelir: kayıpsız.

Her çift 2 bayt: 4 × 2 = **8 bayt**, özgün 16 baytın yarısı. Çözülünce satır aynen geri gelir; bilgi kaybı yoktur.

Adım 6 / 7

```
1 sayaç = 1; renk = ilk piksel
2 sonraki piksel aynı renkse: sayaç = sayaç + 1
3 renk değişince: (sayaç, renk) yaz; sayaç = 1
4 satır bitince: son çifti yaz
▶ boyut: 16 bayt → 4 çift × 2 bayt
6 kötü durum: M B M B …
```

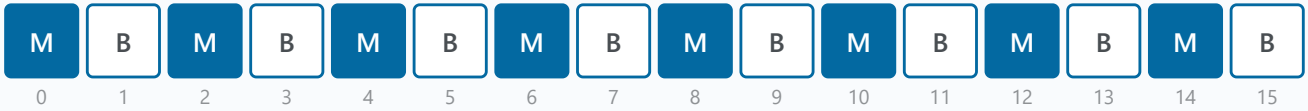
DEĞİŞKENLER

`özgün = 16 bayt`

`sıkıştırılmış = 8 bayt` `oran = %50`

Çalışma Uzunluğu Kodlaması (RLE)

Görüntü satırı (16 piksel, piksel başına 1 bayt)



Çıktı:

16 çift × 2 bayt = 32 bayt: özgün 16 bayttan büyük!

Kötü durum: renk her pikselde değişirse her koşu 1 uzunluktadır; 16 çift = 32 bayt. RLE ancak uzun tekrarlarda (çizim, ikon, tarama) işe yarar.

Adım 7 / 7

```
1 sayaç = 1; renk = ilk piksel
2 sonraki piksel aynı renkse: sayaç = sayaç + 1
3 renk değişince: (sayaç, renk) yaz; sayaç = 1
4 satır bitince: son çifti yaz
5 boyut: 16 bayt → 4 çift × 2 bayt
▶ kötü durum: M B M B …
```

DEĞİŞKENLER

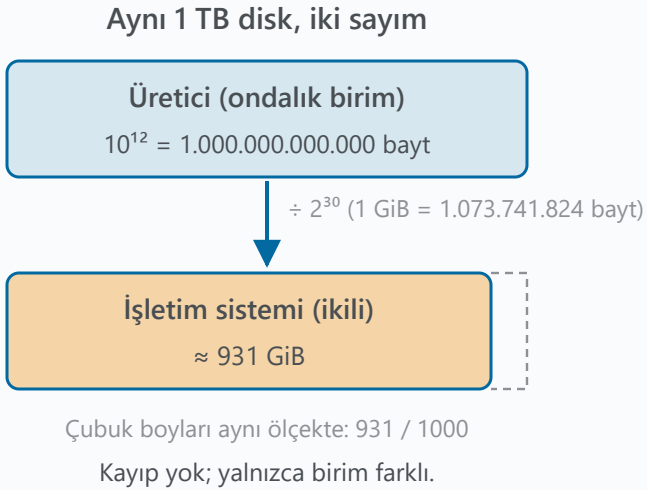
özgün = **16 bayt**

sıkıştırılmış = **32 bayt** oran = **%200**

VERİ TEMSİLİ

Bellek Ölçü Birimleri

Birim	Açılım	İkili karşılığı
bit	Binary digit	0 veya 1
bayt	Byte	8 bit
KB	Kilobyte	1.024 bayt
MB	Megabyte	1.024 KB
GB	Gigabyte	1.024 MB
TB	Terabyte	1.024 GB
PB	Petabyte	1.024 TB



Tablodaki 1.024 katları IEC standardında KiB, MiB, GiB diye yazılır. Disk üreticileri ondalık tanımı kullanır (1 GB = 10^9 bayt); bu yüzden 1 TB disk işletim sisteminde ≈ 931 GiB görünür.

Birimlerin Ölçeği



bit: en küçük birim, 0 ya da 1. Örnek: tek bir anahtar.

Çubuk yükseklikleri ölçekli değildir; her basamakta boyut yaklaşık bin kat büyür.

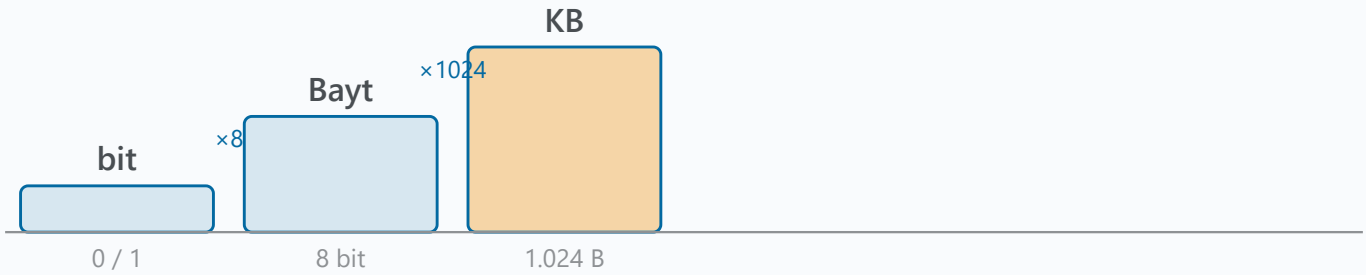
Birimlerin Ölçeği



1 bayt = 8 bit. Bir bayt 2^8 = 256 farklı değer alır; örnek: 1 karakter: “A”.

Çubuk yükseklikleri ölçekli değildir; her basamakta boyut yaklaşık bin kat büyür.

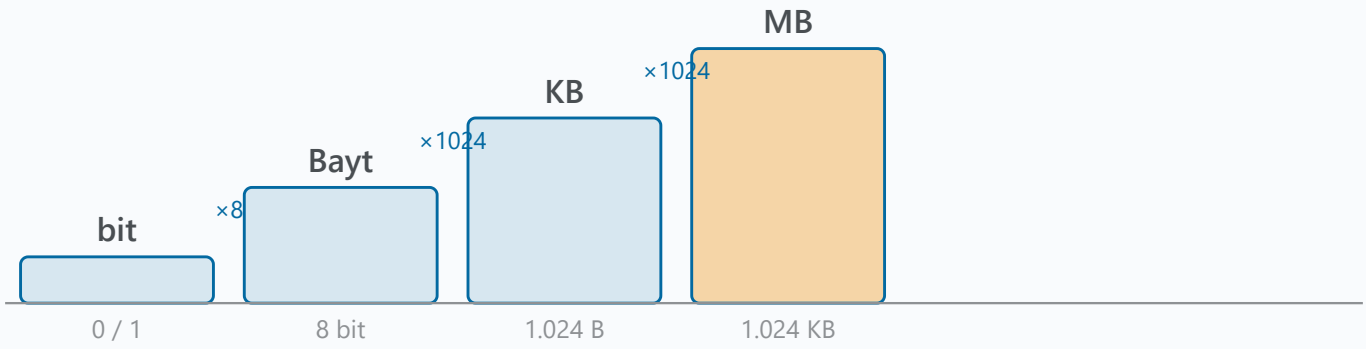
Birimlerin Ölçeği



1 KB = 1.024 B (2^{10} kat). Örnek: yarım sayfa düz metin ≈ 2 KB.

Çubuk yükseklikleri ölçekli değildir; her basamakta boyut yaklaşık bin kat büyür.

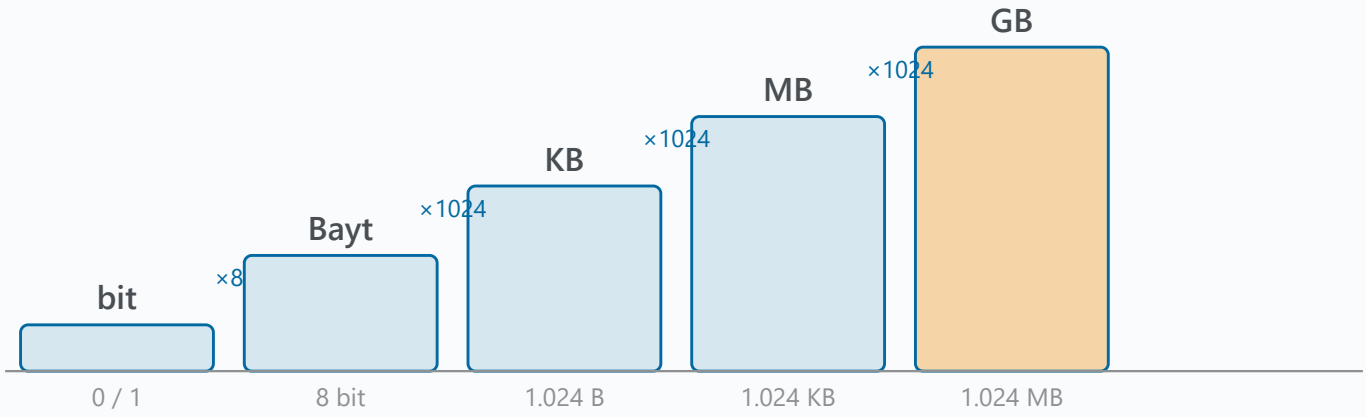
Birimlerin Ölçeği



1 MB = 1.024 KB (2^{10} kat). Örnek: telefon fotoğrafı ≈ 3 MB.

Çubuk yükseklikleri ölçekli değildir; her basamakta boyut yaklaşık bin kat büyür.

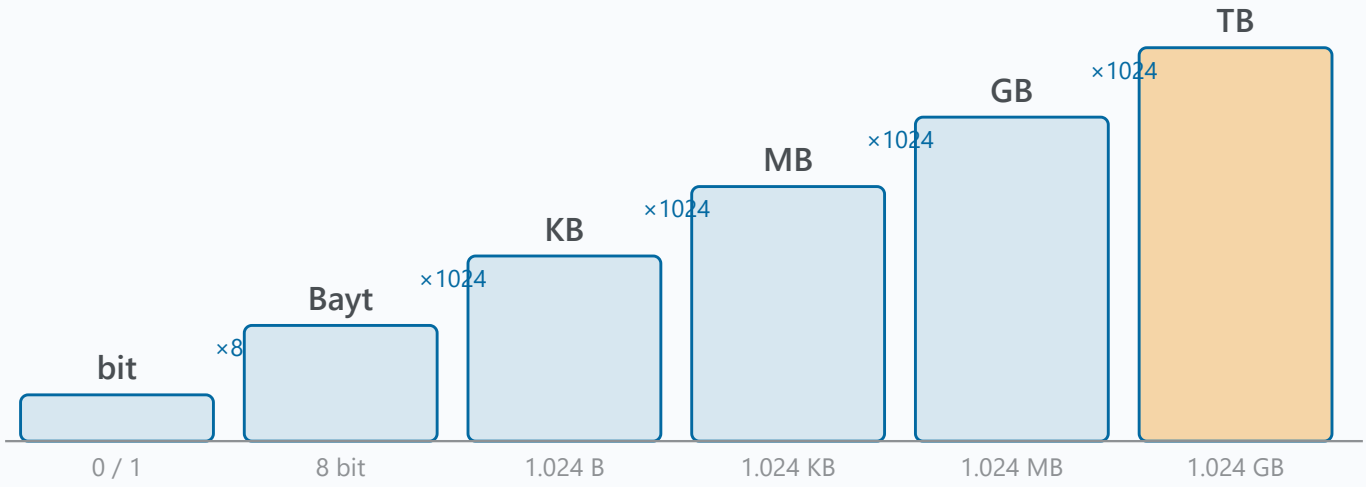
Birimlerin Ölçeği



1 GB = 1.024 MB (2^{10} kat). Örnek: HD film ≈ 4 GB.

Çubuk yükseklikleri ölçekli değildir; her basamakta boyut yaklaşık bin kat büyür.

Birimlerin Ölçeği



1 TB = 1.024 GB (2^{10} kat). Örnek: dizüstü diski ≈ 1 TB.

Çubuk yükseklikleri ölçekli değildir; her basamakta boyut yaklaşık bin kat büyür.

VERİ TEMSİLİ

Bit mi Bayt mı? Hız ve Boyut

Küçük b, büyük B

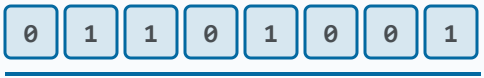
b bit, **B** bayttır: 100 Mb ile 100 MB arasında 8 kat fark vardır.

Hız bit ile ölçülür

Ağ hızı saniyedeki bit sayısıdır: 100 Mbps = 100.000.000 bit/s (M = 10^6).

Boyut bayt ile ölçülür

Dosya ve disk boyutu bayttır: MB, GB.



8 bit (b) = 1 bayt (B)

Hesap: 100 Mbps bağlantıda 3 GB dosya	Sonuç
Hız bayta çevrilir: 100.000.000 ÷ 8	12.500.000 bayt/s = 12,5 MB/s
Dosya bite çevrilir: 3 × 10^9 bayt × 8	24 × 10^9 bit
Süre: 24 × 10^9 ÷ 10^8	240 s = 4 dakika (en iyi durum)

Gerçekte protokol ek yükü ve ağı paylaşan cihazlar nedeniyle süre daha uzundur.

VERİ İŞLEME

Aritmetik ve Mantıksal İşlemler

Aritmetik

Sayısal değerler üzerinde hesaplama yapar.

`+ − × ÷ %`

Karşılaştırma

İki değerin ilişkisini sınar.

`> < ≥ ≤ == !=`

Mantıksal

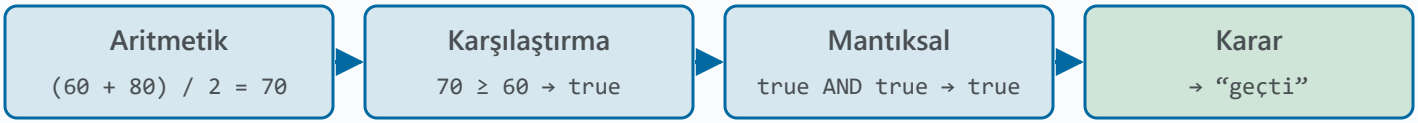
Birden fazla koşulu birleştirir veya tersine çevirir.

`AND OR NOT`

Karar

Sonuca göre programın izleyeceği yolu belirler.

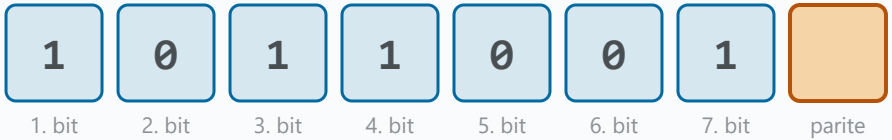
`puan ≥ 60 → geçti`



Örnek: vize 60, final 80, devam_ok = true

Parite Biti ile Hata Yakalama

Gönderici



1'lerin sayısı: 4 → çift

Çift parite kuralı

8 bitteki 1'lerin sayısı
her zaman çift olmalı

tek çıkarsa → hata var
çift çıkarsa → hata yok (?)

Gönderilecek veri **1011001** (ASCII'de "Y", 89). Yedi veri bitinde 4 tane 1 var.

Adım 1 / 5

```
▶ veri = 1011001; 1'leri say
2 çift parite: parite = 0 → 10110010 gönder
3 iletimde 3. bit bozuldu
4 alıcı 1'leri sayar: tekse hata
5 iki bit bozulursa: çift → hata görülmez
```

DEĞİŞKENLER

`veri = 1011001` `1 sayısı = 4` `parite = –`

Parite Biti ile Hata Yakalama

Gönderici

1	0	1	1	0	0	1	0
1. bit	2. bit	3. bit	4. bit	5. bit	6. bit	7. bit	parite

1'lerin sayısı: 4 → çift

Çift parite kuralı

8 bitteki 1'lerin sayısı
her zaman çift olmalı

tek çıkarsa → hata var
çift çıkarsa → hata yok (?)

Çift parite: 1 sayısı zaten çift olduğu için parite biti **0**. Gönderilen 8 bit: 10110010.

Adım 2 / 5

```
1 veri = 1011001; 1'leri say
▶ çift parite: parite = 0 → 10110010 gönder
3 iletimde 3. bit bozuldu
4 alıcı 1'leri sayar: tekse hata
5 iki bit bozulursa: çift → hata görülmez
```

DEĞİŞKENLER

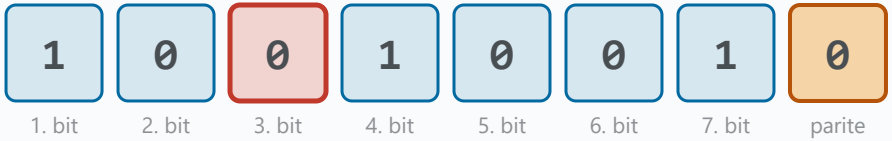
`veri = 1011001` `1 sayısı = 4`

`parite = 0` `gönderilen = 10110010`

VERİ İŞLEME · ADIM ADIM

Parite Biti ile Hata Yakalama

Alıcı



Çift parite kuralı

8 bitteki 1'lerin sayısı
her zaman çift olmalı

tek çıkarsa → hata var
çift çıkarsa → hata yok (?)

İletim sırasında gürültü 3. biti 1'den 0'a çevirdi. Alıcıya **10010010** ulaştı.

Adım 3 / 5

```
1 veri = 1011001; 1'leri say
2 çift parite: parite = 0 → 10110010 gönder
▶ iletimde 3. bit bozuldu
4 alıcı 1'leri sayar: tekse hata
5 iki bit bozulursa: çift → hata görülmez
```

DEĞİŞKENLER

`gönderilen = 10110010`

`alınan = 10010010` `bozuk bit = 1`

Parite Biti ile Hata Yakalama

Alıcı

1. bit	2. bit	3. bit	4. bit	5. bit	6. bit	7. bit	parite
1	0	0	1	0	0	1	0

1'lerin sayısı: 3 → tek

✗ Tek sayı: hata yakalandı → yeniden gönder

Çift parite kuralı

8 bitteki 1'lerin sayısı
her zaman çift olmalı

tek çıkarsa → hata var
çift çıkarsa → hata yok (?)

Alıcı 1'leri sayar: 3, yani **tek**. Kural bozulduğu için hata fark edilir; hangi bitin bozulduğu ise bilinemez, veri yeniden istenir.


```
1 veri = 1011001; 1'leri say
2 çift parite: parite = 0 → 10110010 gönder
3 iletimde 3. bit bozuldu
▶ alıcı 1'leri sayar: tekse hata
5 iki bit bozulursa: çift → hata görülmez
```

DEĞİŞKENLER

`alınan = 10010010` `1 sayısı = 3` `sonuç = hata`

Parite Biti ile Hata Yakalama

Alıcı



1'lerin sayısı: 2 → çift

✓ Çift sayı: hata görülmedi (ama veri bozuk!)

Çift parite kuralı

8 bitteki 1'lerin sayısı
her zaman çift olmalı

tek çıkarsa → hata var
çift çıkarsa → hata yok (?)

İki bit birden bozulursa 1 sayısı yine çift kalır: alıcı veriyi doğru sanar. Parite yalnızca **tek sayıda** bit hatasını yakalar.

Adım 5 / 5

```
1 veri = 1011001; 1'leri say
2 çift parite: parite = 0 → 10110010 gönder
3 iletimde 3. bit bozuldu
4 alıcı 1'leri sayar: tekse hata
▶ iki bit bozulursa: çift → hata görülmez
```

DEĞİŞKENLER

`alınan = 10000010` `1 sayısı = 2`

`sonuç = fark edilmedi`

BİLGİSAYAR TÜRLERİ

Kapasite ve Kullanım Amacına Göre

Süper bilgisayar
Yoğun bilimsel hesaplama

Ana bilgisayar
Çok sayıda güvenilir işlem

Sunucu
Ağ üzerinden hizmet

Kişisel bilgisayar
Bireysel kullanım

Gömülü sistem
Belirli bir cihaz işlevi

İnce istemci
Merkezî sisteme erişim

Mobil cihaz
Taşınabilir kullanım

Kuantum bilgisayar
Özel problem sınıfları

Sınıflandırma kesin sınırlar içermez. Aynı cihaz, kullanım senaryosuna göre birden fazla kategoriyle ilişkilendirilebilir.

BİLGİSAYAR TÜRLERİ

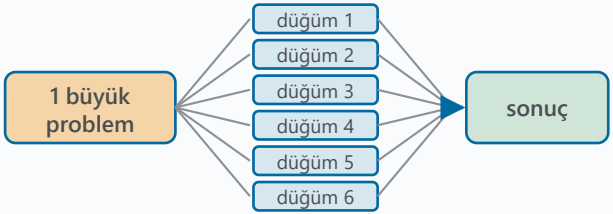
Süper ve Ana Bilgisayarlar

Süper bilgisayar

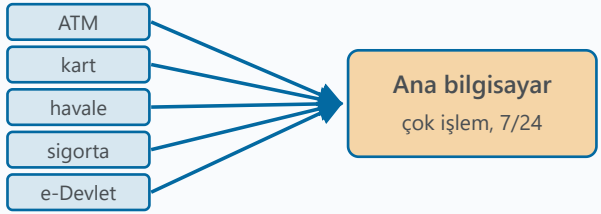
Çok yüksek hesaplama gücü gerektiren bilimsel problemlerde kullanılır.
Örnek: iklim modelleme, fizik simülasyonları, yapay zekâ eğitimi.

Ana bilgisayar

Çok sayıda eş zamanlı ve güvenilir işlemi kesintisiz yürütmek için tasarlanır.
Örnek: bankacılık, sigorta, büyük kamu sistemleri.



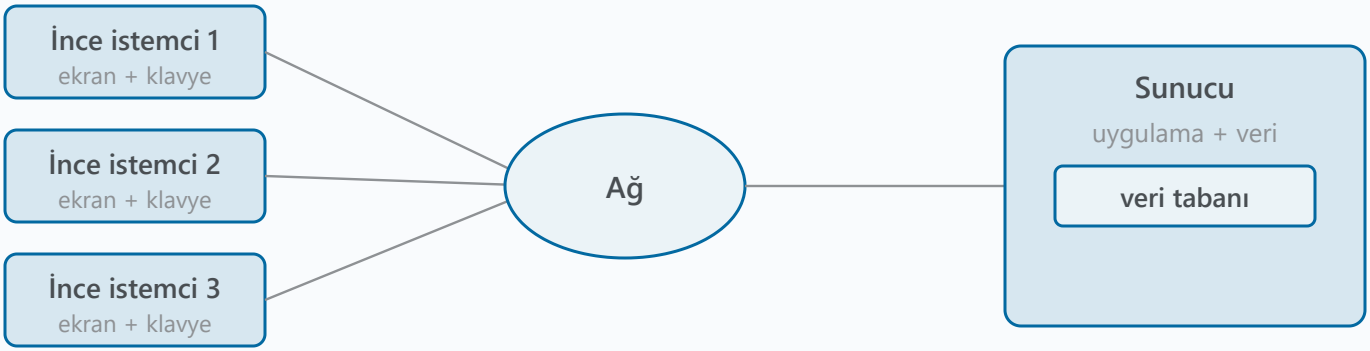
paralel hesaplama → hız



çok sayıda küçük işlem → süreklilik

Öncelik	Süper bilgisayar	Ana bilgisayar
Temel amaç	Hesaplama hızı	İşlem hacmi ve süreklilik
Tipik iş yükü	Büyük bilimsel hesap	Çok sayıda kurumsal işlem

Sunucular ve İnce İstemciler



Birçok istemci ağ üzerinden aynı **sunucuya** bağlanır.

Adım 1 / 4

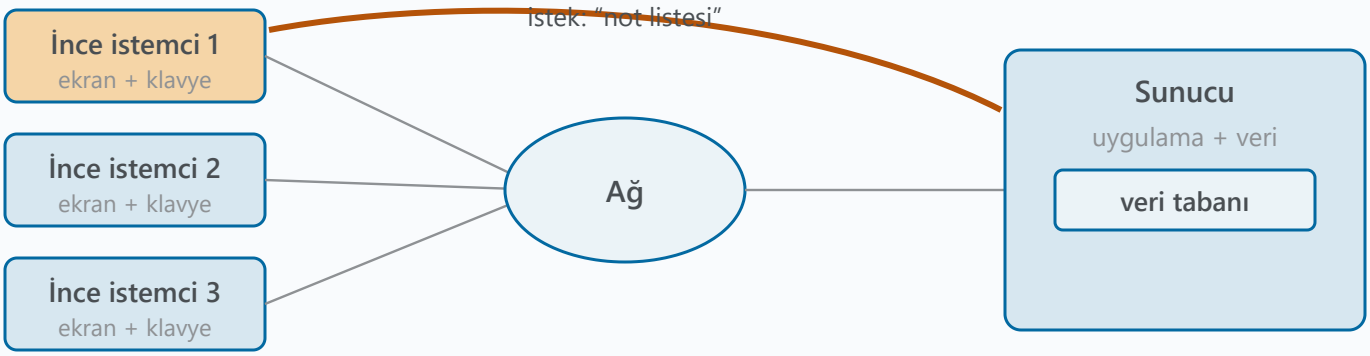
Sunucu

Web, e-posta, dosya, veri tabanı veya uygulama hizmetlerini ağ üzerinden sunar.

İnce istemci

İşlemin çoğunu sunucuya bırakır; yerel donanım ve yönetim ihtiyacı azalır.

Sunucular ve İnce İstemciler



İstek: İstemci 1 not listesini ister; istek ağ üzerinden sunucuya gider.

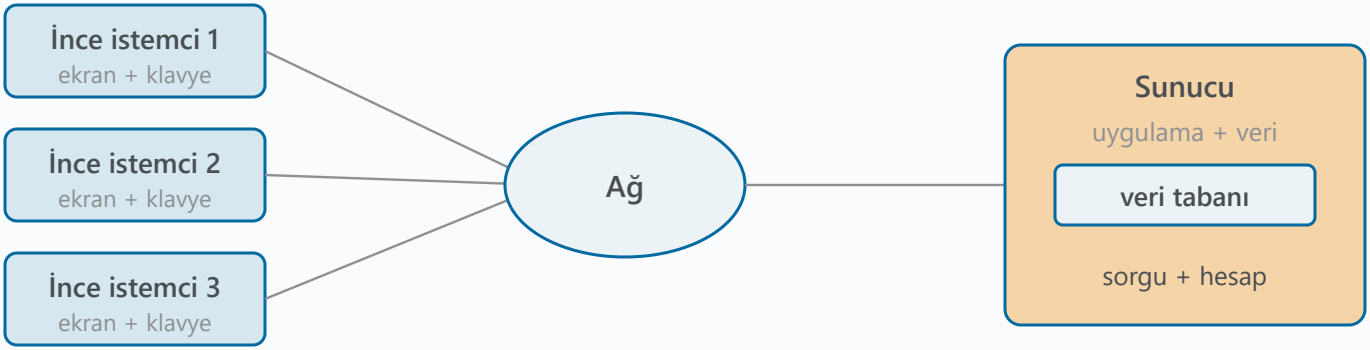
Sunucu

Web, e-posta, dosya, veri tabanı veya uygulama hizmetlerini ağ üzerinden sunar.

İnce istemci

İşlemin çoğunu sunucuya bırakır; yerel donanım ve yönetim ihtiyacı azalır.

Sunucular ve İnce İstemciler



İşlem: Asıl iş sunucuda yapılır: veri tabanı sorgulanır, sonuç hesaplanır.

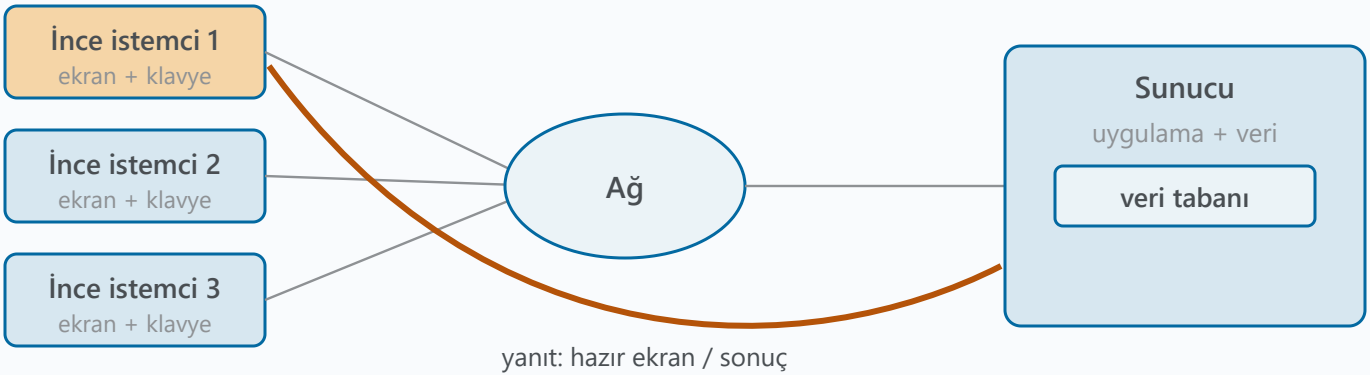
Sunucu

Web, e-posta, dosya, veri tabanı veya uygulama hizmetlerini ağ üzerinden sunar.

İnce istemci

İşlemin çoğunu sunucuya bırakır; yerel donanım ve yönetim ihtiyacı azalır.

Sunucular ve İnce İstemciler



Yanıt: Sonuç istemciye döner. İnce istemci yalnızca gösterir ve girdi alır.

Sunucu

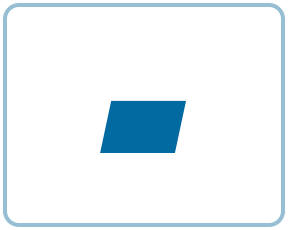
Web, e-posta, dosya, veri tabanı veya uygulama hizmetlerini ağ üzerinden sunar.

İnce istemci

İşlemin çoğunu sunucuya bırakır; yerel donanım ve yönetim ihtiyacı azalır.

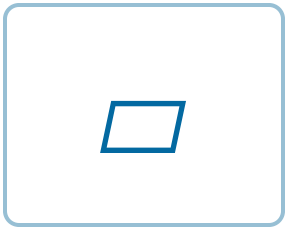
BİLGİSAYAR TÜRLERİ

Kişisel ve Mobil Cihazlar



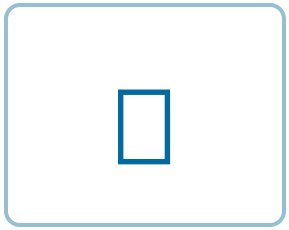
Masaüstü

Sabit kullanım, genişletilebilir donanım



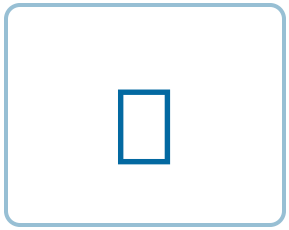
Dizüstü

Taşınabilir yapı, batarya



Tablet

Dokunmatik ekran, hafif kullanım



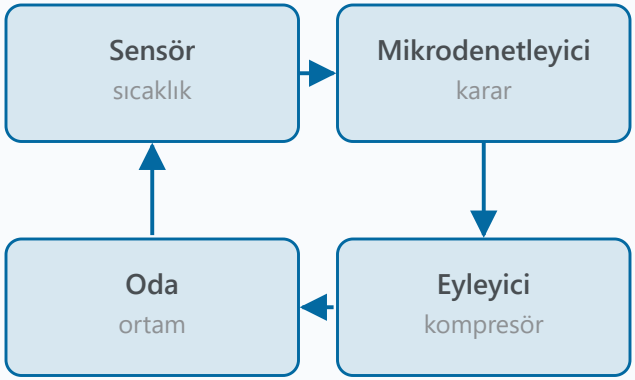
Telefon

Mobil bağlantı, sensörler, uygulamalar

Cihaz seçiminde performans kadar taşınabilirlik, enerji tüketimi, ekran boyutu ve kullanım amacı da önemlidir.

Gömülü Bilgisayarlar

Gömülü bilgisayar, daha büyük bir cihazın içinde belirli bir görevi yerine getirir.



hedef sıcaklık: 24 °C

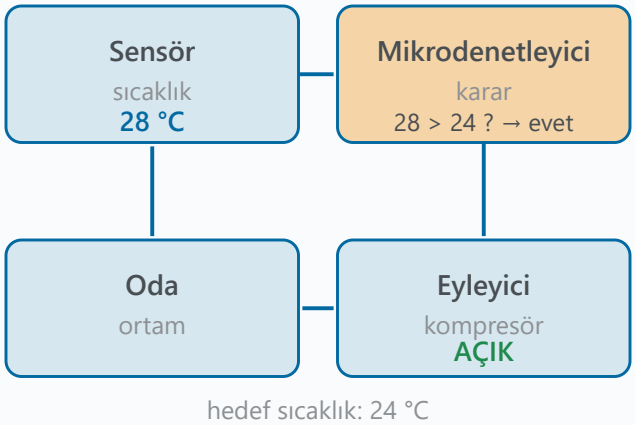
Klimadaki gömülü bilgisayar sürekli bir **ölç → karar ver → uygula** döngüsü çalıştırır.

Adım 1 / 4

- **Örnekler:** otomobil motor denetimi, çamaşır makinesi, klima, tıbbi ölçüm cihazı.
- Genellikle düşük güç tüketir, sınırlı kaynakla çalışır.
- Güvenilirlik ve gerçek zamanlı tepki önemlidir.
- Kullanıcı, içinde bir bilgisayar olduğunu fark etmeyebilir.

Gömülü Bilgisayarlar

Gömülü bilgisayar, daha büyük bir cihazın içinde belirli bir görevi yerine getirir.



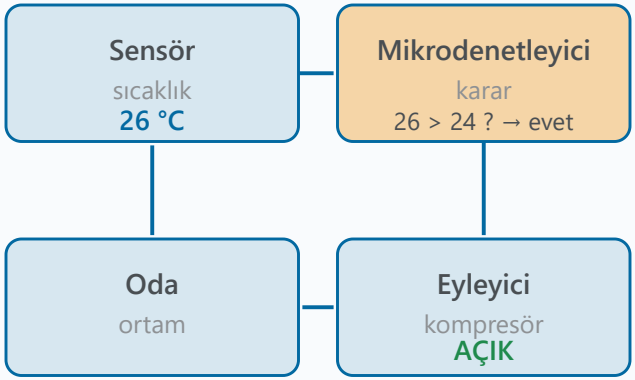
Sensör **28 °C** ölçer. 28 > 24 → kompresör **açık**; oda soğur.

Adım 2 / 4

- **Örnekler:** otomobil motor denetimi, çamaşır makinesi, klima, tıbbi ölçüm cihazı.
- Genellikle düşük güç tüketir, sınırlı kaynakla çalışır.
- Güvenilirlik ve gerçek zamanlı tepki önemlidir.
- Kullanıcı, içinde bir bilgisayar olduğunu fark etmeyebilir.

Gömülü Bilgisayarlar

Gömülü bilgisayar, daha büyük bir cihazın içinde belirli bir görevi yerine getirir.

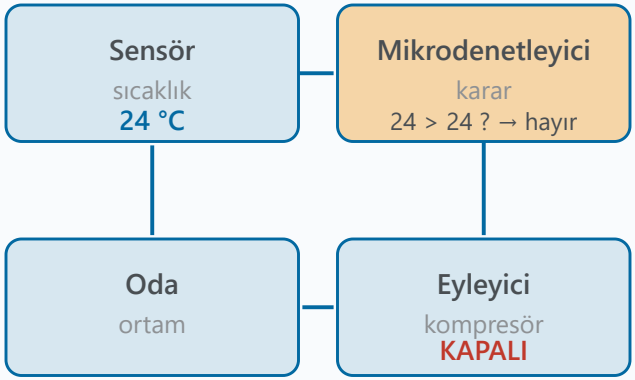


Sensör **26 °C** ölçer. 26 > 24 → kompresör **açık**; oda soğur.

- **Örnekler:** otomobil motor denetimi, çamaşır makinesi, klima, tıbbi ölçüm cihazı.
- Genellikle düşük güç tüketir, sınırlı kaynakla çalışır.
- Güvenilirlik ve gerçek zamanlı tepki önemlidir.
- Kullanıcı, içinde bir bilgisayar olduğunu fark etmeyebilir.

Gömülü Bilgisayarlar

Gömülü bilgisayar, daha büyük bir cihazın içinde belirli bir görevi yerine getirir.



hedef sıcaklık: 24 °C

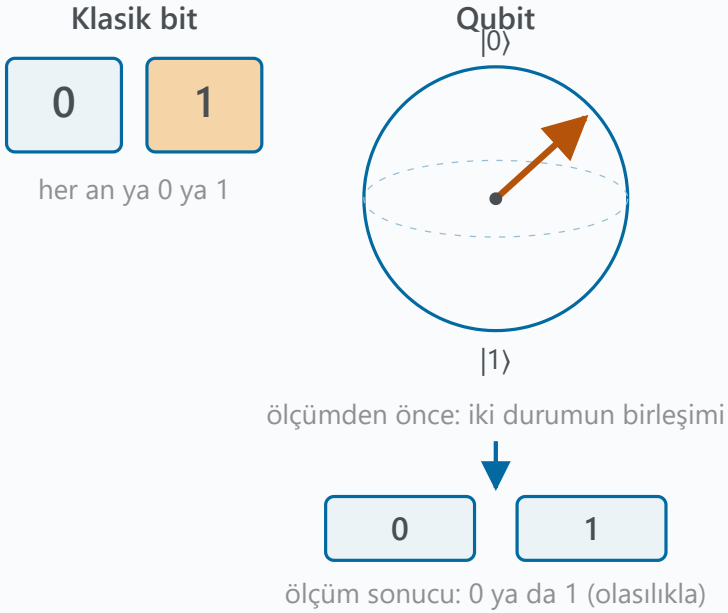
Sensör **24 °C** ölçer. 24 > 24 yanlış → kompresör **kapanır**. Döngü ölçmeye devam eder.

- **Örnekler:** otomobil motor denetimi, çamaşır makinesi, klima, tıbbi ölçüm cihazı.
- Genellikle düşük güç tüketir, sınırlı kaynakla çalışır.
- Güvenilirlik ve gerçek zamanlı tepki önemlidir.
- Kullanıcı, içinde bir bilgisayar olduğunu fark etmeyebilir.

Kuantum Bilgisayarlar

Kuantum bilgisayarlar bilgiyi **qubit** adı verilen kuantum bitleriyle işler.

- Belirli problemlerde farklı hesaplama yöntemleri sunar.
- Kriptografi, optimizasyon, kimya ve malzeme araştırmalarında incelenir.
- Genel amaçlı günlük kullanımda klasik bilgisayarların yerini almış değildir.



BİLGİ TEKNOLOJİLERİ

Bilgi Teknolojisi (BT)

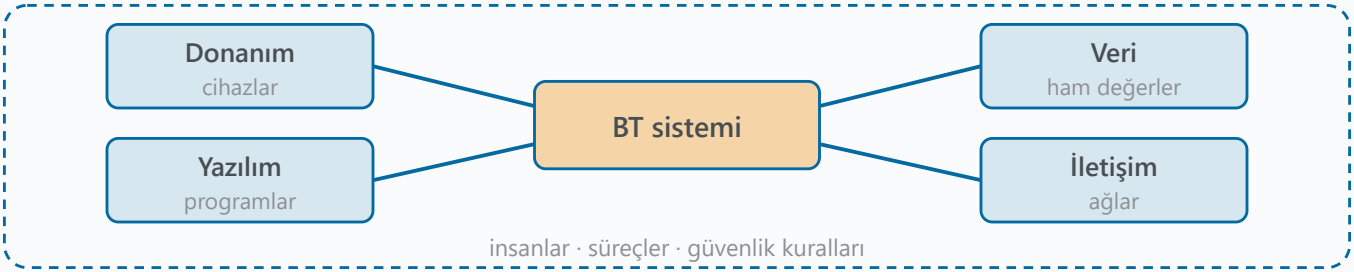
Bilgi teknolojisi, veriyi toplamak, işlemek, saklamak ve iletmek için kullanılan sistemlerin bütünüdür.

Donanım
Fiziksel cihazlar

Yazılım
Programlar ve işletim sistemleri

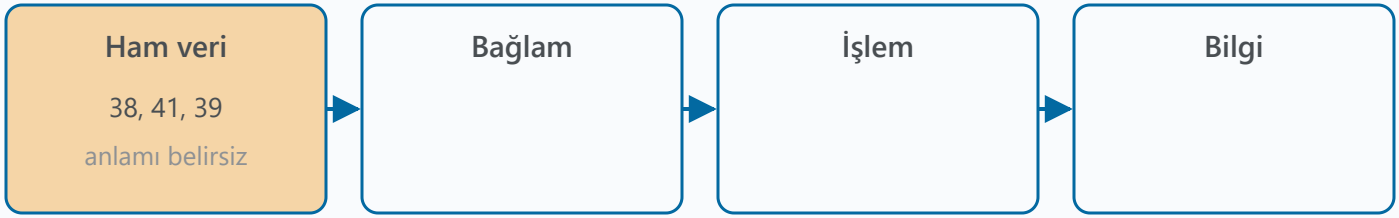
Veri
İşlenen ham değerler

İletişim
Ağlar ve veri aktarımı



BT sistemi yalnızca bilgisayardan oluşmaz. İnsanlar, süreçler ve güvenlik kuralları da sistemin çalışmasını etkiler.

Veri ve Bilgi Arasındaki Fark



38, 41, 39: yalnızca sayılar. Ne ölçüldüğü bilinmedikçe anlamı yoktur.

Adım 1 / 4

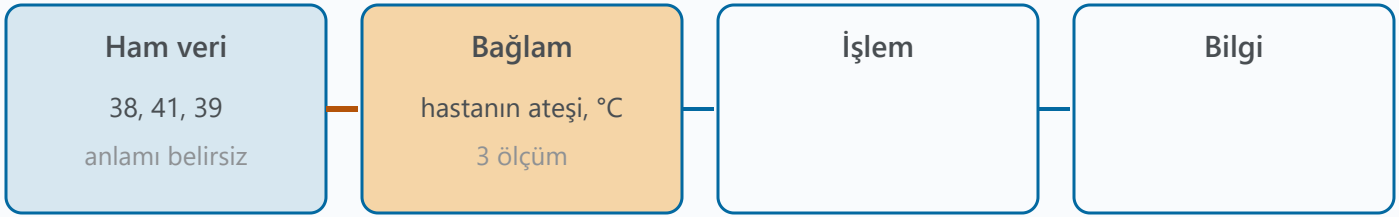
Veri

Ham ölçüm, sayı, metin veya kayıttır. Bağlam olmadan anlamı belirsiz kalabilir.

Bilgi

Verinin düzenlenmesi ve yorumlanmasıyla ortaya çıkan anlamlı sonuçtur.

Veri ve Bilgi Arasındaki Fark



Bağlam eklenir: bunlar bir hastanın üç ateş ölçümüdür (°C).

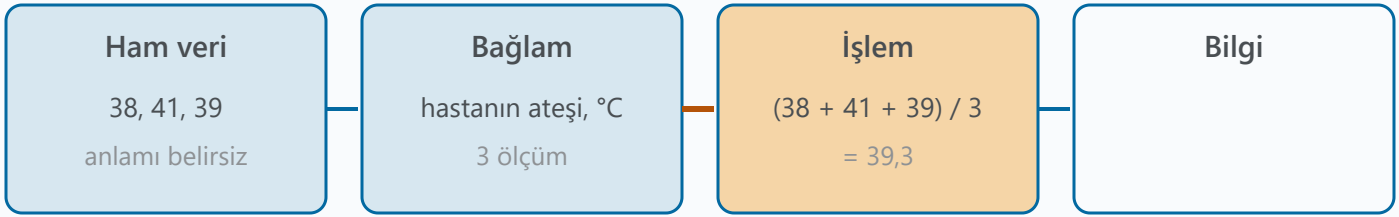
Veri

Ham ölçüm, sayı, metin veya kayıttır. Bağlam olmadan anlamı belirsiz kalabilir.

Bilgi

Verinin düzenlenmesi ve yorumlanmasıyla ortaya çıkan anlamlı sonuçtur.

Veri ve Bilgi Arasındaki Fark



İşlem: ortalama = (38 + 41 + 39) / 3 = **39,3**.

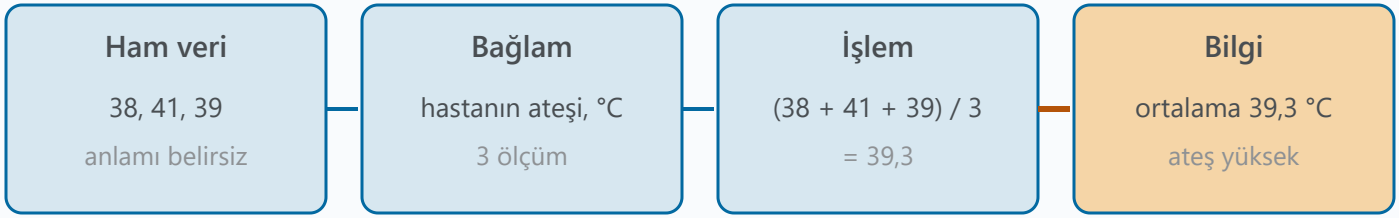
Veri

Ham ölçüm, sayı, metin veya kayıttır. Bağlam olmadan anlamı belirsiz kalabilir.

Bilgi

Verinin düzenlenmesi ve yorumlanmasıyla ortaya çıkan anlamlı sonuçtur.

Veri ve Bilgi Arasındaki Fark



Bilgi: "Ortalama ateş 39,3 °C, yüksek". Artık karar verilebilir.

Veri

Ham ölçüm, sayı, metin veya kayıttır. Bağlam olmadan anlamı belirsiz kalabilir.

Bilgi

Verinin düzenlenmesi ve yorumlanmasıyla ortaya çıkan anlamlı sonuçtur.

BİLGİ TEKNOLOJİLERİ

BT'nin Günlük Yaşamdaki Rolü

Eğitim

Çevrim içi dersler, dijital kaynaklar ve öğrenme platformları.

Sağlık

Elektronik kayıtlar, görüntüleme sistemleri ve hasta takibi.

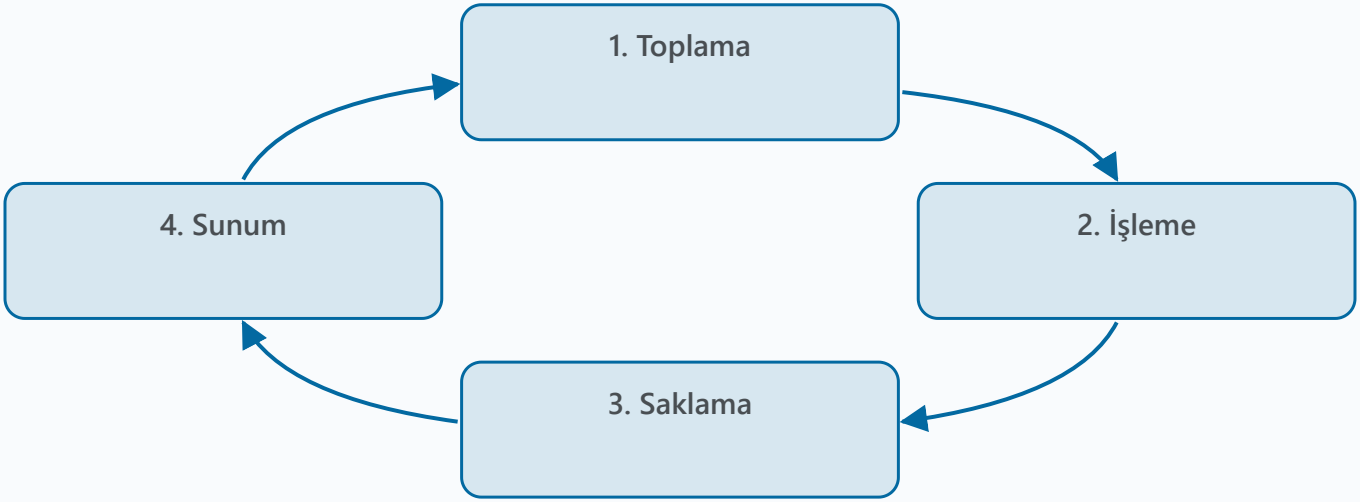
Finans

Bankacılık, ödeme sistemleri ve işlem güvenliği.

İş ve eğlence

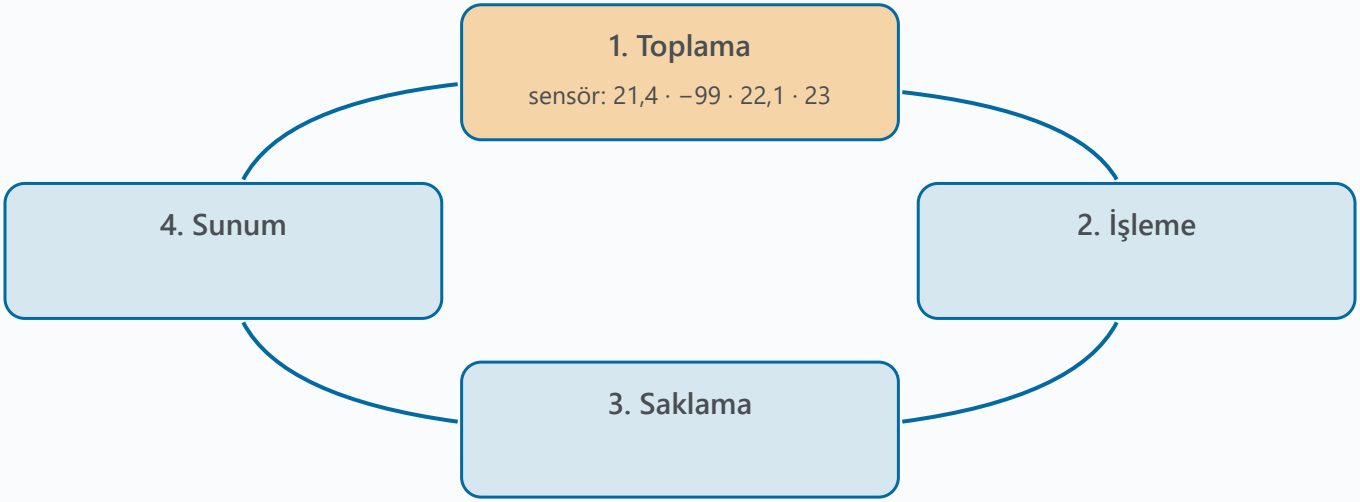
Kurumsal iletişim, otomasyon, oyun ve medya hizmetleri.

Veri Yaşam Döngüsü



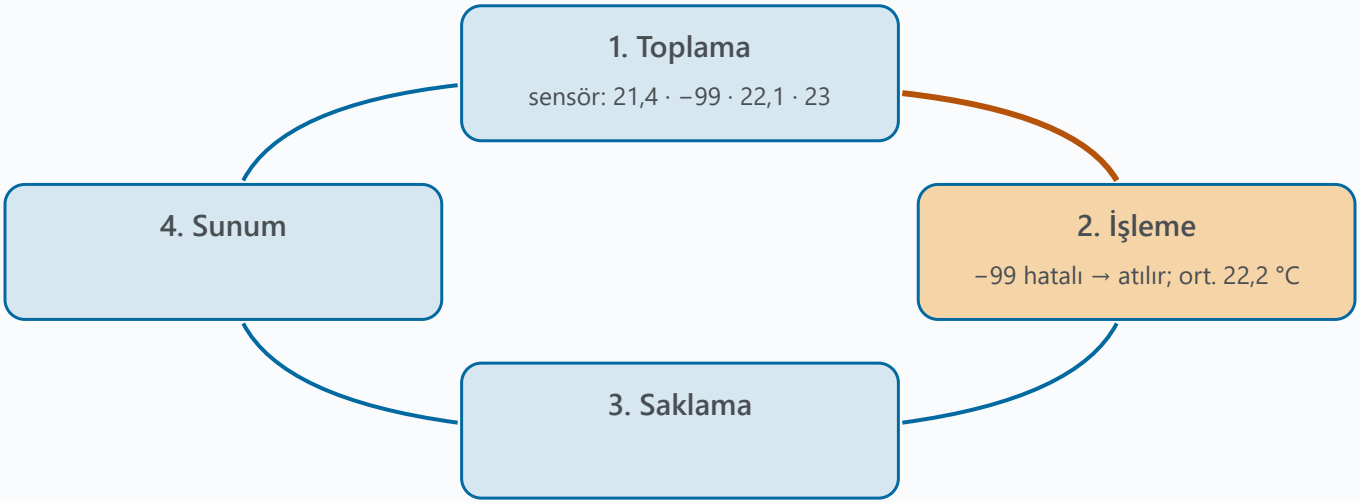
Örnek: bir hava istasyonunun verisi dört aşamadan geçer ve döngü yeniden başlar.

Veri Yaşam Döngüsü



Toplama: Veri insanlardan, makinelerden veya sensörlerden alınır. Sensör 4 ölçüm gönderdi.

Veri Yaşam Döngüsü



İşleme: Hatalar ayıklanır, hesap yapılır. −99 (arızalı ölçüm) atılır; ortalama 22,2 °C.

Veri Yaşam Döngüsü



Saklama: Veri dosyada, veri tabanında veya bulutta korunur.

Veri Yaşam Döngüsü



Sunum: Sonuç rapor, tablo, grafik veya uygulama ekranıyla paylaşılır; yeni kararlar yeni veri toplamayı başlatır.

BİLGİ TEKNOLOJİLERİ

Avantajlar ve Sorumluluklar

Hız ve otomasyon

Veriler hızlı işlenir, tekrarlanan görevler otomatikleşir.

Erişim ve iletişim

Bilgiye farklı yerlerden ulaşmak ve paylaşmak kolaylaşır.

Mahremiyet ve güvenlik

Kişisel veriler yetkisiz erişime ve kayba karşı korunmalıdır.

Doğruluk ve etik

Yanlış veri ve önyargılı sistemler hatalı sonuç üretebilir.

Teknolojik kolaylık, veriyi doğru, güvenli ve sorumlu kullanma yükümlülüğünü ortadan kaldırmaz.

BİLGİ TEKNOLOJİLERİ

Bilgi Teknolojilerinin Geleceği

Yapay zekâ

Veriden örüntü öğrenen karar destek ve üretim sistemleri.

Nesnelerin interneti

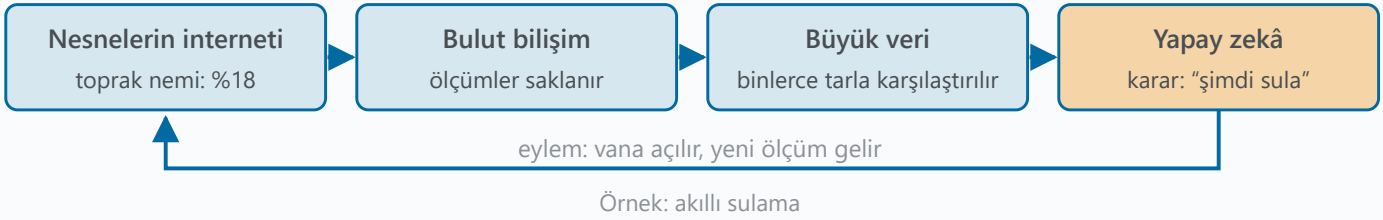
Sensörlü cihazların ağ üzerinden veri paylaşması.

Bulut bilişim

Hesaplama ve depolama kaynaklarının ağ üzerinden sağlanması.

Büyük veri analitiği

Yüksek hacimli ve çeşitli verilerden anlamlı sonuç çıkarılması.



Örnek: akıllı sulama

Yeni teknolojiler kadar enerji kullanımı, erişilebilirlik, güvenlik ve etik de önem kazanır.

ÖZET

Bölüm Özeti

- Bilgisayarlar veriyi giriş, işlem, çıkış ve depolama döngüsünde işler.
- Tüm veri tipleri bilgisayarda bit dizileriyle temsil edilir.
- Bit sayısı arttıkça temsil edilebilen değer sayısı iki katına çıkar; ikili–onluk dönüşüm basamak değerleriyle yapılır.
- AND, OR ve NOT koşulları birleştirir; toplama ve karşılaştırma ALU'da yapılır.
- Ses örnekleme, nicemleme ve kodlamayla sayıya dönüşür; sıkıştırma kayıpsız (ZIP, PNG) ya da kayıplı (JPEG, MP3) olabilir.
- Parite biti tek bitlik iletim hatalarını yakalar; ağ hızı bit/s, dosya boyutu bayt ile ölçülür.
- Bilgisayar türlerini kullanım amacı, kapasite ve çalışma ortamı ayırır.
- Bilgi teknolojileri donanım, yazılım, veri, iletişim, insanlar ve süreçlerden oluşur.

Bilgisayar veriyi işler. İnsan, bağlam ekleyerek onu bilgiye dönüştürür.

Kontrol Soruları

1. Bir ATM işlemindeki giriş, işlem, çıkış ve depolama adımları nelerdir?
2. 8 bit neden 256 farklı değer gösterebilir?
3. RAM ile SSD arasındaki temel fark nedir?
4. Sunucu ile ana bilgisayar hangi yönlerden ayrılır?
5. Bir sıcaklık ölçümünün bilgiye dönüşmesi için hangi bağlam gerekir?
6. Bilgi teknolojilerinde güvenlik neden yalnızca teknik bir konu değildir?

ÖZET

Alıştırmalar

1. 37 sayısını ikiliye çevirin (2'ye bölme yöntemiyle).
2. 101011_2 sayısı onluk sistemde kaçtır?
3. Türk alfabesinin yalnızca 29 büyük harfini kodlamak için en az kaç bit gerekir? Kaç kod boşta kalır?
4. “Hi” kelimesini ASCII kodlarıyla ve 8 bitlik baytlar olarak yazın (H = 72, i = 105).
5. 1920×1080 piksellik, piksel başına 3 bayt kullanan bir görüntü sıkıştırılmadan kaç bayt tutar? Yaklaşık kaç MB eder?
6. 30 fps hızındaki 10 saniyelik bir videoda kaç kare vardır?

▶ **Cevapları göster**

Kaynaklar

- Brookshear ve Brylow, *Computer Science: An Overview*, Pearson.
- Stallings, *Computer Organization and Architecture*, Pearson.
- Unicode Consortium, *The Unicode Standard*.
- NIST, *Glossary of Key Information Security Terms*.

Son

Temel Kavramlar

Sonraki sunum: **Bilgisayar Donanımı** · **Çalışma soruları** · Simülatör: **Sayı Tabanı Dönüştürücü** · **Terim sözlüğü**