

Sayı Sistemleri

Günlük hayatta 10'luk sayı sistemini kullanıyoruz. 10'luk sistemde bir sayının değeri aslında her bir basamak değerinin 10 sayısının ilgili kuvvetiyle çarpımının toplanmasıyla elde edilir.

Örneğin $1273 = (3 * 1) + (7 * 10) + (2 * 100) + (1 * 1000)$

Ancak bilgisayar sistemlerinde bütün bilgiler ikilik sistemde ifade edilir.

Genel olarak sayı sistemi kaçlıksa o sayı sisteminde o kadar sembol bulunur.

Örneğin 10'luk sistemde 10 adet sembol vardır ve bu semboller 0, 1, 2, 3, 4, 5, 6, 7, 8, 9'dur.

Aynı şekilde ikilik sayı sisteminde yalnızca iki adet sembol bulunur. Yani yalnızca 0 ve 1.

Bir sayıyı başka bir sayı sisteminde ifade etmek o sayının değerini değiştirmez. Yalnızca sayının gösteriliş biçimi değişir. Sayısal değeri 2 olan büyüklüğü çeşitli farklı sayı sistemlerinde farklı biçimlerde gösterebiliriz ama sayının büyüklüğünü değiştirmiş olmayız.

İkilik sistemde her bir basamağa 1 bit denir. Bit kelimesi **binary digit** sözcüklerinden türetilmiştir.

Örneğin 1011 sayısı 4 bittir. (Ya da 4 bit uzunluğundadır).

11011001 sayısı 8 bittir.

8 bitlik bir büyüklük bir byte olarak isimlendirilir.

1 kilobyte 1K = 1024 byte dır. (yani 2^{10} byte)

1 mega byte 1 MB = 1024 Kilo byte dır. (yani 2^{20} byte)

1 gigabyte 1 GB = 1024 MB dır. (yani 2^{30} byte)

1 terabyte 1 TB = 1024 GB dır. (yani 2^{40} byte)

1 petabyte 1PB = 1024 TB dır. (yani 2^{50} byte)

1 exabyte 1EB = 1024 PB dır. (yani 2^{60} byte)

1 zettabyte 1ZB = 1024 EB dir.(yani 2^{70} byte)

1 yottabyte 1YB = 1024 ZB dır.(yani 2^{80} byte)

Kilo büyüklük olarak 1000 kat anlamına gelmektedir, ancak bilgisayar alanında Kilo 2'nin 1000'e en yakın kuvveti olan 2^{10} yani 1024 kat olarak kullanılır.

4 bit 1 Nybble (Nibble şeklinde de yazılır)

8 bit 1 byte

16 bit 1 word

32 bit 1 double word

64 bit 1 quadro word

olarak da isimlendirilmektedir.

8 bitlik bir alana yazılacak en büyük sayı nedir?

$1111\ 1111 = 255$ dir. $(1 + 2 + 4 + 8 + 16 + 32 + 64 + 128 = 255)$

8 bitlik bir alana yazılacak en küçük sayı nedir?
0000 0000 = 0'dır.

İkilik sisteme ilişkin genel işlemler

i. İkilik sistemdeki bir sayının 10'luk sistemde ifade edilmesi:
ikilik sayı sisteminde ifade edilen bir sayının 10'luk sistemdeki karşılığını hesaplamak için en sağdan başlayarak bütün basamakları tek tek 2'nin artan kuvvetleriyle çarpılır. Örneğin :

$$1\ 0\ 1\ 1 = 1 * 2^0 + 1 * 2^1 + 0 * 2^2 + 1 * 2^3 = 11$$

$$0010\ 1001 = (1 * 1) + (1 * 8) + (1 * 32) = 41$$

ii. 10'luk sistemdeki bir sayının 2'lik sistemde ifadesi:

Sayı sürekli olarak 2 ye bölünür. Her bölümden kalan oluşturulacak sayı sırasıyla sağdan sola yerleştirilir.

İkinci bir yöntem ise 10'luk sistemde ifade edilen sayıdan sürekli olarak 2'nin en büyük kuvvetini çıkarmaktır. 2'nin çıkarılan her bir kuvveti için ilgili basamağa 1 değeri yazılır.

iii. İkilik sistemde ifade edilen bir sayının 1'e tümleyeni. Sayının tüm bitlerinin tersinin alınmasıyla elde edilir. Yani sayıdaki 1'ler 0 ve 0'lar 1 yapılır.

Bir sayının 1'e tümleyeninin 1'e tümleyeni sayının yine kendisidir.

iv. İkilik sistemde ifade edilen bir sayının 2'ye tümleyeninin bulunması:

Önce sayının 1'e tümleyeni yukarıdaki gibi bulunur. Daha sonra elde edilen sayıya 1 eklenirse sayının 2'ye tümleyeni bulunmuş olur. 2'ye tümleyeni bulmak için daha daha pratik bir yol daha vardır: sayının en sağından başlayarak ilk defa 1 görene kadar (ilk görülen 1 dahil) sayının aynısı yazılır, daha sonraki tüm basamaklar için basamağın tersi yazılır. (Yani 1 için 0 ve 0 için 1) Örneğin:

1110 0100 sayısının ikiye tümleyeni 0001 1100 dır.

0101 1000 sayısının ikiye tümleyeni 1010 1000 dır.

Bir sayının ikiye tümleyeninin ikiye tümleyeni sayının kendisidir.

Negatif bir tamsayı ikilik sistemde nasıl gösterilir?

Negatif tamsayıların da ifade edildiği ikilik sayı sistemine işaretli ikilik sayı sistemi (*signed binary system*) denir. İşaretli ikilik sayı sisteminde, negatif sayıları göstermek için hemen hemen tüm bilgisayar sisteminde aşağıdaki yol izlenir:

Sayının en soldaki biti, yani en yüksek anlamlı biti (*most significant bit*) işaret biti (*sign bit*) olarak kabul edilir. Ve bu bit **1** ise sayı negatif, bu bit **0** ise sayı pozitif olarak değerlendirilir. İkilik sistemde bir negatif sayı aynı değerdeki pozitif sayının ikiye tümleyenidir. Örnek olarak, ikilik sistemde yazacağımız -27 sayısı yine ikilik sistemde yazılan 27 sayısının ikiye tümleyenidir.

Pozitif olan sayıların değerini tıpkı işaretsiz sayı sisteminde olduğu gibi elde ederiz:

0001 1110 işaretli sistemde pozitif bir sayıdır. (Decimal olarak 30 sayısına eşittir.)

Ancak negatif olan sayıların değerini ancak bir dönüşümle elde edebiliriz:

1001 1101 işaretli sistemde negatif bir sayıdır. (Çünkü işaret biti 1)

2'lik sistemde ifade edilen negatif bir sayının 10'luk sistemde hangi negatif sayıya eşit olduğunu nasıl bulunur?

Sayının en soldaki biti (the most significant bit) işaret bitidir. Bu bit 1 ise sayı negatiftir. Sayının kaç eşi olduğunu hesaplamak için ilk önce sayının 2'ye tümleyeni bulunur. Ve bu sayının hangi pozitif sayıya karşılık geldiğini hesap edilir. Elde etmek istenen sayı, bulunan pozitif sayı ile aynı değerdeki negatif sayı olacaktır.

Örneğin 1001 1101 sayısının 10'luk sistemde hangi sayıya karşılık geldiği bulunmak istenirse:

Sayının en soldaki biti 1 olduğuna göre bu sayı negatif bir sayı olacaktır. Hangi negatif sayı olduğunu bulmak için sayının 2'ye tümleyenini alınır.

1001 1101 sayısının ikiye tümleyeni 0110 0011 sayıdır.

Bu sayının 10'luk sistemde hangi sayıya denk olduğu hesaplanırsa:

$$(1 * 1 + 1 * 2 + 0 * 4 + 0 * 8 + 0 * 16 + 1 * 32 + 1 * 64 = 99)$$

İlk yazılan sayının -99 olduğu anlaşılmış olur.

10'luk sistemde ifade edilen negatif sayıların işaretli ikilik sistemde yazılması:

Önce sayının aynı değerli fakat pozitif olanı ikilik sistemde ifade edilir: Daha sonra yazılan sayının ikiye tümleyenini alınarak, yazmak istenilen sayı elde edilir.

Örnek: İkilik sistemde -17 yazmak istenirse;

önce 17 yazılır.

0001 0001

bu sayının 2'ye tümleyeni alınır

1110 1111 sayısı elde edilir.

Sayı değeri aynı olan Negatif ve Pozitif sayılar birbirlerinin ikiye tümleyenleridir.

İkilik sistemde gösterilmiş olsa da aynı sayının negatifıyla pozitifinin toplamı 0 değerini verecektir. (Deneyiniz!)

Bir byte'lık (8 bitlik) bir alana yazabileceğimiz (işaret bitini dikkate almadan) en büyük sayı 255 (1111 1111) ve en küçük sayı ise 0'dır.(0000 0000). Peki işaret biti dikkate alındığında 1 byte'lık alana yazılabilecek en büyük ve en küçük sayılar ne olabilir?

En büyük sayı kolayca hesaplanabilir. İşaret biti 0 olacak (yani sayı pozitif olacak) ve sayı değerini en büyük hale getirmek için diğer bütün bit değerleri 1 olacak, bu sayı 0111 1111 sayıdır. Bu sayıyı desimal sisteme dönüştürsek 127 olduğunu görürüz. Peki ya en küçük negatif sayı kaçtır ve nasıl ifade edilir?

0111 1111 sayısının ikiye tümleyenini alındığında -127 sayısını elde edilir.
1000 0001 (127) Bu sayıdan hala 1 çıkartabilir.
1000 0000 (-128) 1 byte alana yazılabilecek en küçük negatif sayıdır.

Burada dikkat edilmesi gereken iki önemli nokta vardır:

1 byte alana yazılabilecek en büyük sayı sınırı aşıldığında negatif bölgeye geçilir.

0111 1111 (en büyük pozitif tamsayı)

1 (1 toplarsak)

1000 0000 (-128 yani en küçük tamsayı)

yani 1 byte alana yazılabilecek en büyük tamsayıya 1 eklendiğinde 1 byte alana yazılabilecek en küçük tamsayıyı elde ederiz.

1 byte alana yazılabilecek en küçük tamsayıdan 1 çıkardığımızda da 1 byte alana yazılabilecek en büyük tamsayıyı elde ederiz.

16'lık sayı sistemi (hexadecimal system) ve 8'lik sayı sistemi (octal system)

Bilgisayarların tamamen 2'lik sistemde çalıştığını söylemiştik, ama yukarıda görüldüğü gibi 2'lik sistemde sayıların ifade edilmesi hem çok uzun hem de zahmetli. Bu yüzden, yazım ve algılama kolaylığı sağlamak için 16'lık ve 8'lik sayı sistemleri de kullanılmaktadır.

16'lık ve 8'lik sayı sistemlerinde sayılar daha yoğun olarak kodlanıp kullanılabilir.

Başta da söz edildiği gibi 10'luk sistemde 10, 2'lik sistemde ise 2 sembol bulunmaktadır. Bu durumda 16'lık sayı sisteminde de 16 sembol bulunur.

İlk 10 sembol 10'luk sistemde kullanılan sembollerle tamamen aynıdır :

1, 2, 3, 4, 5, 6, 7, 8, 9,

Daha sonraki semboller

A = 10

B = 11

C = 12

D = 13

E = 14

F = 15

16'lık sayı sisteminde yazılmış bir sayıyı 10'luk sisteme çevirmek için, en sağdan başlayarak basamak değerleri 16'nın artan kuvvetleriyle çarpılır :

$$01AF = (15 * 1) + (10 * 16) + (1 * 256) + (0 * 4096) = 431$$

10'luk sistemde yazılmış bir sayıyı 16'lık sisteme çevirmek için 10 luk sistemden 2'lik sisteme yapılan dönüşümlerdekine benzer şekilde sayı sürekli 16 ya bölünerek, kalanlar soldan sağa doğru yazılır.

Pratikte 16 lık sayı sistemlerinin getirdiği önemli bir avantaj vardır. Bu avantaj 16 lık sayı sistemi ile 2'lik sayı sistemi arasındaki dönüşümlerin kolay bir şekilde yapılmasıdır.

16'lık sistemdeki her digit 2'lik sistemdeki 4 bit (1 Nibble) alan ile ifade edilebilir :

0000	0
0001	1
0010	2
0011	3
0100	4
0101	5
0110	6
0111	7
1000	8
1001	9
1010	A
1011	B
1100	C
1101	D
1110	E
1111	F

Örnek : 2ADF Hex sayısının (en sondaki H sayının hexadecimal olarak gösterildiğini anlatır yani sayıya ilişkin bir sembol değildir) 16'lık sistemde ifadesi :

2 = 0010
A = 1010
D = 1101
F = 1111

Bu durumda 2ADFH = 0010 1010 1101 1111

2'lik sistemden 16'lık sisteme yapılacak dönüşümler de benzer şekilde yapılabilir :

Önce sayıları sağdan başlayarak dörder dörder ayırırız (en son dört eksik kalırsa sıfır ile tamamlarız.)
Sonra her bir dördlük grup için doğrudan Hexadecimal karşılığını yazarız.

1010 1110 1011 0001 = AEB1H
0010 1101 0011 1110 = 2D3EH

Soru : hexadecimal sayı sisteminde 2 byte'lık bir alanda yazılmış olan 81AC H sayısı pozitif mi negatif midir?

Cevap : Sayının yüksek anlamlı biti 1 olduğu için, işaretli sayı sistemlerinde sayı negatif olarak değerlendirilecektir. (1000 0001 1010 1100)

8'lik sayı sistemi

Daha az kullanılan bir sayı sistemidir.

8 adet sembol vardır. (0 1 2 3 4 5 6 7)

8'lik sayı sisteminin her bir digiti 2'lik sistemde 3 bit ile ifade edilir.

000	0
001	1
010	2
011	3
100	4
101	5
110	6
111	7

8'lik sayı sisteminin de kullanılma nedeni, 2'lik sayı sistemine göre daha yoğun bir ifade tarzı olması, ve ikilik sayı sistemiyle, 8'lik sayı sistemi arasında yapılacak dönüşümlerin çok kolay bir biçimde yapılabilmesidir.

GERÇEK SAYILARIN BELLEKTE TUTULMASI

Sistemlerin çoğu gerçek sayıları **IEEE 754** standardına göre tutarlar. (*Institute of Electrical and Electronics Engineers*) Bu standarda göre gerçek sayılar için iki ayrı format belirlenmiştir:

single precision format (tek hassasiyetli gerçek sayı formatı)

Bu formatta gerçek sayı 32 bit (4 byte) ile ifade edilir.
32 bit üç ayrı kısma ayrılmıştır.

1. İşaret biti (sign bit) (1 bit)
Aşağıda S harfi ile gösterilmiştir.
İşaret biti 1 ise sayı negatif, işaret biti 0 ise sayı pozitiftir.

2. Üstel kısım (exponent) (8 bit)
Aşağıda E harfleriyle gösterilmiştir.

3. Ondalık kısım (fraction) (23 bit)
Aşağıda F harfleriyle gösterilmiştir.

S	EEEEEEEE	FFFFFFFFFFFFFFFFFFFFFFF
31	30-----23	22-----0

Aşağıdaki formüle göre sayının değeri hesaplanabilir :

V sayının değeri olmak üzere:

E = 255 ise ve F 0 dışı bir değer ise V = NaN (Not a number) bir gerçek sayı olarak kabul edilmez.
Örnek :

0 11111111 000010000001000000000000	= Sayı değil
1 11111111 00010101010001001010101	= Sayı değil

E = 255 ise ve F = 0 ise ve S = 1 ise V = -sonsuz
E = 255 ise ve F = 0 ise ve S = 1 ise V = +sonsuz

$0 < E < 255$ ise

$$V = (-1)^S * 2^{(E-127)} * (1.F)$$

Önce sayının fraction kısmının başına 1. eklenir. Daha sonra bu sayı $2^{(E-127)}$ ile çarpılarak noktanın yeri ayarlanır. Noktadan sonraki kısım 2'nin artan negatif kuvvetleriyle çarpılarak elde edilecektir.

Örnekler :

$$\begin{aligned} 0\ 10000000\ 000000000000000000000000 &= +1 * 2^{(128-127)} * 1.0 \\ &= 2 * 1.0 \\ &= 10.00 \\ &= 2 \end{aligned}$$

$$\begin{aligned} 0\ 10000001\ 101000000000000000000000 &= +1 * 2^{(129-127)} * 1.101 \\ &= 2^2 * 1.101 \\ &= 110.100000 \\ &= 6.5 \end{aligned}$$

$$\begin{aligned} 1\ 10000001\ 101000000000000000000000 &= -1 * 2^{(129-127)} * 1.101 \\ &= -2^2 * 1.101 \\ &= -110.100000 \\ &= -6.5 \end{aligned}$$

$$\begin{aligned} 0\ 00000001\ 000000000000000000000000 &= +1 * 2^{(1-127)} * 1.0 \\ &= 2^{-126} \end{aligned}$$

E = 0 ve F sıfır dışı bir değer ise

$$V = (-1)^S * 2^{(-126)} * (0.F)$$

Örnekler :

$$\begin{aligned} 0\ 00000000\ 100000000000000000000000 &= +1 * 2^{-126} * 0.1 \\ &= \end{aligned}$$

$$\begin{aligned} 0\ 00000000\ 000000000000000000000001 &= +1 * 2^{-149} * 0.000000000000000000000001 \\ &= 2^{-149} \text{ (en küçük pozitif değer)} \end{aligned}$$

E = 0 ve F = 0 ve S = 1 ise V = -0

E = 0 ve F = 0 ve S = 0 ise V = 0

double precision format (çift hassasiyetli gerçek sayı formatı)

Bu formatta gerçek sayı 64 bit (8 byte) ile ifade edilir.
64 bit üç ayrı kısma ayrılmıştır.

1. İşaret biti (sign bit) (1 bit)
Aşağıda S harfi ile gösterilmiştir.
İşaret biti 1 ise sayı negatif, işaret biti 0 ise sayı pozitiftir.

2. Üstel kısım (exponent) (11 bit)
Aşağıda E harfleriyle gösterilmiştir.

3. Ondalık kısım (fraction) (52 bit)
Aşağıda F harfleriyle gösterilmiştir.

S EEEEEEEEEEE FF
63 62-----52 51-----0

Aşağıdaki formüle göre sayının değeri hesaplanabilir :

Aşağıdaki formüle göre sayının değeri hesaplanabilir :

V sayının değeri olmak üzere:

E = 2047 ise ve F 0 dışı bir değer ise V = NaN (Not a number) bir gerçek sayı olarak kabul edilmez.

E = 2047 ise ve F = 0 ise ve S = 1 ise V = -sonsuz

E = 2047 ise ve F = 0 ise ve S = 1 ise V = +sonsuz

$0 < E < 2047$ ise

$$V = (-1)^S * 2^{(E-1023)} * (1.F)$$

Önce sayının fraction kısmının başına 1. eklenir. Daha sonra bu sayı $2^{(E-1023)}$ ile çarpılarak noktanın yeri ayarlanır. Noktadan sonraki kısım 2'nin artan negatif kuvvetleriyle çarpılarak elde edilecektir.

E = 0 ve F sıfır dışı bir değer ise

$$V = (-1)^S * 2^{(-126)} * (0.F)$$

E = 0 ve F = 0 ve S = 1 ise V = -0

E = 0 ve F = 0 ve S = 0 ise V = 0