



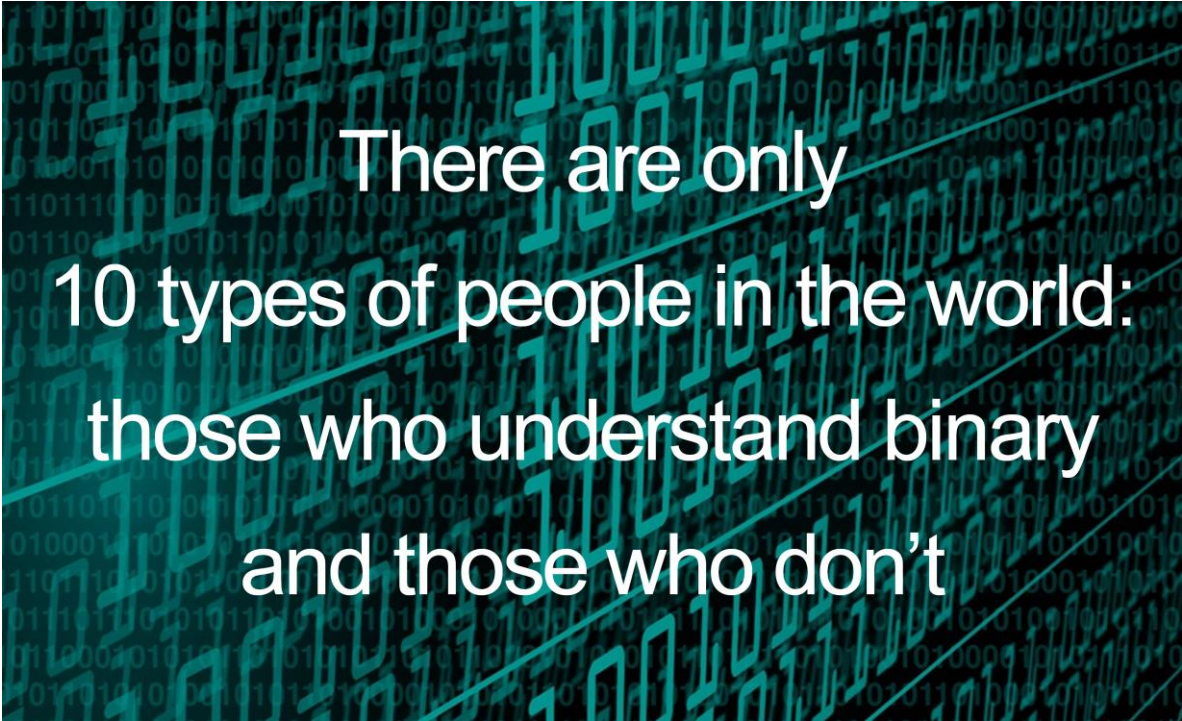
UNIVERSITAT
POLITÀCNICA
DE VALÈNCIA



Data storage and representation

The Binary System

In this unit we'll learn the basics about the binary system.



There are only
10 types of people in the world:
those who understand binary
and those who don't

Let's begin with a typical geek joke: there are 10 types of people: those who understand binary and those who don't. Do you get it? If **you** don't, don't worry. I'll try to explain it.

decimal	binary
0	0
1	1

$2^1 = 2$ possibilities

with one bit, we can just represent two possibilities: zero and one.

decimal	binary
0	00
1	01
2	10
3	11

$2^2 = 4$ possibilities

using two bits, and with all possible combinations, we can represent 4 situations. That's two raised to two.

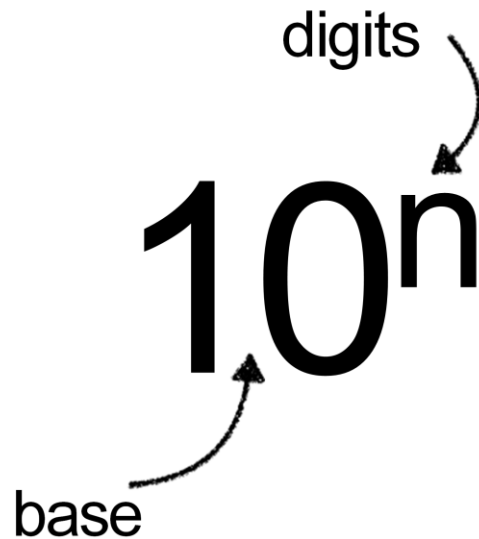
decimal	binary
0	000
1	001
2	010
3	011
4	100
5	101
6	110
7	111

$2^3 = 8$ possibilities

5

and we can continue with 3 bits, resulting 8 possibilities, from zero zero zero to one one one.

#digits	#combinations
0	1
1	10 (0-9)
2	100 (0-99)
3	1000 (0-999)


 10^n

6

In the decimal system we know how to calculate that: the number of different possibilities that we can count is the numeric base (10) raised to the number of digits that we use. For example, with 3 digits we can count from zero to nine hundred and ninety nine. That's ten raised to 3 numbers.

#digits	#combinations
0	1
1	2
2	4
3	8
4	16
5	32
6	64
7	128
8	256
9	512
10	1024

The diagram shows the mathematical expression 2^n . A curved arrow points from the word "base" to the number 2. Another curved arrow points from the word "digits" to the superscript n.

7

The equivalent in binary is that we have two (our numeric base) raised to the digits that we use. In the table on the left you'll find the first 10 powers of two. Are they familiar? Exactly, the memory, or the capacity of a usb pen is measured with these figures.

10^3	10^2	10^1	10^0
2	1	9	3

$$2 \times 1000 + 1 \times 100 + 9 \times 10 + 3 \times 1 = 2193$$

8

How can we convert a binary number into a decimal one? **It's** simple. Think about the meaning of a decimal number. The figures represent the units, ten, hundred, and so on. So we just multiply the figure for the correspondent power of ten and sum the result

10^3	10^2	10^1	10^0
2	1	9	3

$$2 \times 1000 + 1 \times 100 + 9 \times 10 + 3 \times 1 = 2193$$

2^3	2^2	2^1	2^0
1	1	0	1

$$1 \times 8 + 1 \times 4 + 0 \times 2 + 1 \times 1 = 13$$

9

The binary system is exactly the same: we have to multiply each figure by the power of two that corresponds to its position and sum the result. In this case, eight plus four plus one, that is, thirteen.

represent 167 in binary

2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
128	64	32	16	8	4	2	1
1							

$$137-128 = 39$$

10

The opposite operation is simple too. To represent a number in binary, let's say one hundred and sixty seven, we have to locate the next smaller power of two: one hundred and twenty-eight. We write down a one in this position, we subtract the value to the original number and continue doing the same. Now, we have to represent thirty-nine

represent 167 in binary

2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
128	64	32	16	8	4	2	1
1							
1	0	1					

$$137 - 128 = 39$$

$$39 - 32 = 7$$

We fill with zeros the positions not used. We do the same: mark thirty-two, the difference is seven, and we continue...

represent 167 in binary

2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
128	64	32	16	8	4	2	1
1							
1	0	1					
1	0	1	0	0	1		

$$137 - 128 = 39$$

$$39 - 32 = 7$$

$$7 - 4 = 3$$

four

represent 167 in binary

2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
128	64	32	16	8	4	2	1
1							
1	0	1					
1	0	1	0	0	1		
1	0	1	0	0	1	1	

$$137-128 = 39$$

$$39-32 = 7$$

$$7 - 4 = 3$$

$$3 - 2 = 1$$

Two

represent 167 in binary

2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0	
128	64	32	16	8	4	2	1	
1								$137-128 = 39$
1	0	1						$39-32 = 7$
1	0	1	0	0	1			$7 - 4 = 3$
1	0	1	0	0	1	1		$3 - 2 = 1$
1	0	1	0	0	1	1	1	$1 - 1 = 0$

14

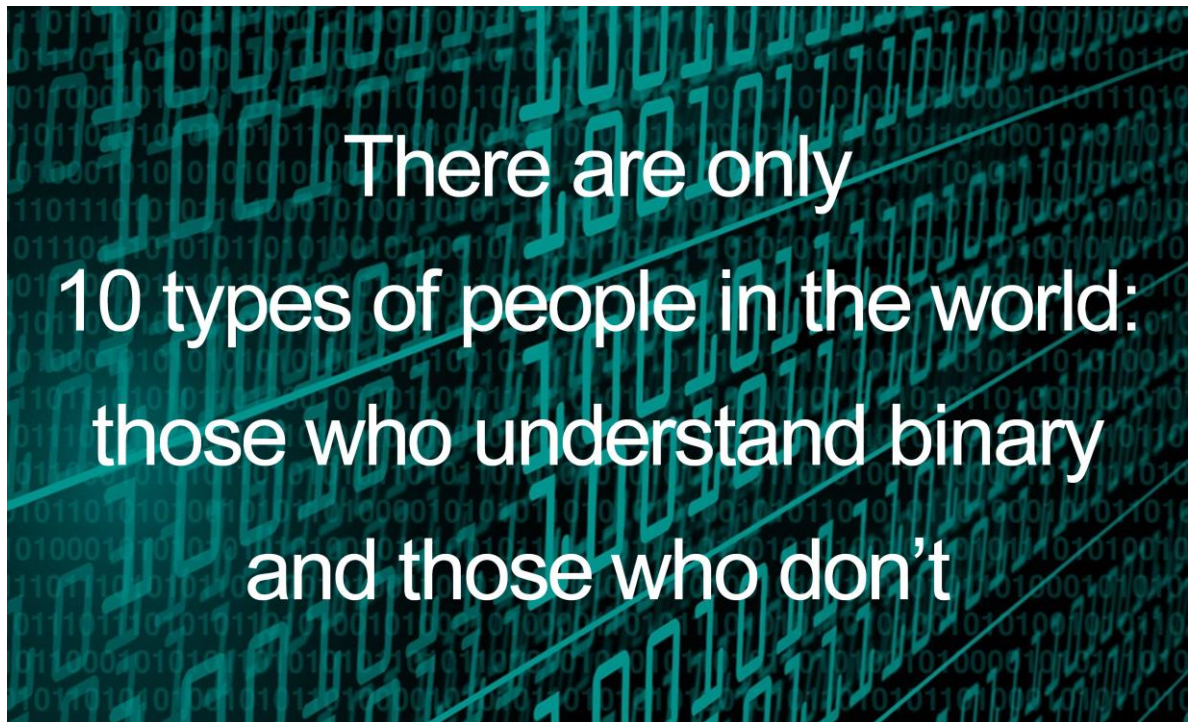
and one

represent 167 in binary

2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0	
128	64	32	16	8	4	2	1	
1								$137-128 = 39$
1	0	1						$39-32 = 7$
1	0	1	0	0	1			$7 - 4 = 3$
1	0	1	0	0	1	1		$3 - 2 = 1$
1	0	1	0	0	1	1	1	$1 - 1 = 0$
$1 \times 128 + 1 \times 32 + 1 \times 4 + 1 \times 2 + 1 \times 1$								

15

So this is the representation of one hundred and sixty seven. You can check that it's correct.



And maybe you can now get the joke: there are only TWO types of people in the world.

decimal	binary	octal
0	000 000	0
1	000 001	1
2	000 010	2
3	000 011	3
4	000 100	4
5	000 101	5
6	000 110	6
7	000 111	7
8	001 000	10
9	001 001	11
10	001 010	12

3-bit
combination

The binary system is very useful to convert to other numeric system useful in computing. For example, the octal system, that uses 8 digits, has a direct translation if we take the bits in groups of three.

unix file permission

owner	group	others	
rwX	rwX	rwX	r: read w: write x: execute

symbolic				numeric
rwXrwXrwX	111	111	111	777
rw-r-r--	110	100	100	644

18

Octal representation is used to represent the file permissions in Unix, where we have three permissions: read, write and execution. That can be codified by an octal number.

decimal	Binary	hexadecimal
0	0000 0000	0
1	0000 0001	1
2	0000 0010	2
3	0000 0011	3
4	0000 0100	4
5	0000 0101	5
6	0000 0110	6
7	0000 0111	7
8	0000 1000	8
9	0000 1001	9
10	0000 1010	A
11	0000 1011	B
12	0000 1100	C
13	0000 1101	D
14	0000 1110	E
15	0000 1111	F

4-bit
combination

Or the hexadecimal system, that uses base sixteen, is equivalent to take the bits in groups of four.

MAC address

identifier for network controllers

48 bits = 6 bytes = 6 groups of 2 hexadecimal numbers

The diagram illustrates the conversion of the first two hexadecimal digits of a MAC address, '01:23', into their binary representation. The text '01:23:45:67:89:ab' is shown at the top. Two curved arrows point from the '01' and '23' to the binary strings '0000 0001' and '1010 1011' respectively, which are positioned below the first two pairs of hexadecimal digits.

01:23:45:67:89:ab

0000 0001 1010 1011

20

Hexadecimal codes are used for long chains of bits, such as the MAC address of a network interface card, formed by forty-eight bits...

0000ff	005fff	008fff	00a8ff	00bfff	00ffff	50ffff	5fdfff	5fa8ff	5fbfff	5f5fff	5f00ff
21	27	33	39	45	51	87	81	75	69	63	57
RGB color code											
8700ff	875fff	8787ff	87a8ff	87d7ff	87ffff	afffff	afd7ff	afa8ff	af87ff	af5fff	af00ff
93	99	105	111	117	123	159	153	147	141	135	129
8700d7	875fd7	8787d7	87afd7	87d7d7	87ffd7	afffd7	afd7d7	afa8d7	af87d7	af5fd7	af00d7
92	98	104	110	116	122	158	152	146	140	134	128
8700af	875faf	8787af	87afaf	87d7af	87ffa8	afffa8	afd7af	afa8af	af87af	af5faf	af00af
91	97	103	109	115	121	157	151	145	139	133	127
870087	875f87	878787	87af87	87d787	87ff87	afff87	afd787	afa887	af8787	af5f87	af0087
90	96	102	108	114	120	156	150	144	138	132	126
87005f	875f5f	87875f	87af5f	87d75f	87ff5f	afff5f	afd75f	afa85f	af875f	af5f5f	af005f
89	95	101	107	113	119	155	149	143	137	131	125
870000	875f00	878700	87af00	87d700	87ff00	afff00	afd700	afa800	af8700	af5f00	af0000
88	94	100	106	112	118	154	148	142	136	130	124
d70000	d75f00	d78700	d7af00	d7d700	d7ff00	ffff00	ffd700	ffa800	ff8700	ff5f00	ff0000

or the representation of colours in RGB, using two hexadecimal figures for each component: red, green and blue

Attribution

The sources of some of these figures are :

- <https://www.flickr.com/photos/x1brett/6665955101>
- https://en.wikipedia.org/wiki/DIP_switch#/media/File:DIP_switch_01_Pengo.jpg
- https://commons.wikimedia.org/wiki/File:Xterm_color_chart.png