

Forum:[Hack Forums](#)

Thread: [Learn to calculate binary code \[Tutorial\] Easy](#)

ShadowReaperZ - November 17, 2017, 7:19 am

Intro

Hey there members,

I am kind of new to the forums, but I enjoy it so far.
Today I want to share some knowledge of my side.
Please don't mind my bad English spelling.
Good luck!

1. foreknowledge

The first thing you need to know is that 1 byte has 8 bits.
So the next time you see a image or text included with an example like this:
01011001000110010101110111011001110110010001100011011010
You have to cut it in 1 byte (8 bits)
Example: 01011001 00011001 01011101 11011001 11011001 00011000 11011010

2. Make the standard table

(8 bit system)

The first thing you want to do, is creating the standard table.

It goes like this: 128-64-32-16-8-4-2-1.

You may think why those random numbers?

Well actually it is not so random. Because $128+64= 192$. And if you add all those numbers together you get 255.

And of course we know that 192 is an IP address (192.168)

And 255 is the subnet mask. (255.255.255.0)

So the max number a byte can contain is 255 by a 8 bit system.

$$(1 \cdot 2^3) + (0 \cdot 2^2) + (1 \cdot 2^1) + (1 \cdot 2^0) = 8 + 0 + 2 + 1 = 11$$

You can see that in binary numbers bit holds the value of increasing powers of 2. That makes counting in binary pretty easy. Starting at zero and going through 20 counting in decimal and binary looks like this:

0 =	0
1 =	1
2 =	10
3 =	11
4 =	100
5 =	101
6 =	110
7 =	111
8 =	1000
9 =	1001

10 = 1010
11 = 1011
12 = 1100
13 = 1101
14 = 1110
15 = 1111
16 = 10000
17 = 10001
18 = 10010
19 = 10011
20 = 10100

3. Convert numbers to bytes

Make sure you have your standard table by hand: 128-64-32-16-8-4-2-1

As you know by now 1 byte contains 8 bits with the maximum number of 255. (11111111)

Lets start with converting numbers to bits with the example: 155

As you can see the number is an odd. All the numbers with an odd always end with 1 on the end of the byte.

1.Does 128 fit in 155? Yes = 1 / No = 0 | 1

- Now we made our first bit : 1 because 128 fits in 155 so it's positive (1)

- We have 27 left because 155-128 is 27.

- Our byte looks now like this (1)

2.Does 64 fit in 27? Yes = 1 / No = 0 | 0

- As you can see 64 doesn't fit in 27 so it's negative (0)

- Our byte looks now like this= (10)

3.Does 32 fit in 27? Yes = 1 / No = 0 | 0

- Our byte looks now like this= (100)

4.Does 16 fit in 27? Yes = 1 / No = 0 | 1

- This is our second positive number in the byte so (1)

- Our byte looks now like this=(1001)

- We have 11 left because 27-16= 11

5.Does 8 fit in 11? Yes = 1 / No = 0 | 1

- Now we have only 3 left

- Our byte looks now like this=(10011)

6.Does 4 fit in 3? Yes = 1 / No = 0 | 0

- Our byte looks now like this=(100110)

7. Does 2 fit in 3? Yes = 1 / No = 0 | 1

- Our byte looks now like this=(1001101)

8. Does 1 fit in 1? Yes = 1 / No = 0 | 1

- Our byte looks now like this=(10011011)

Now our byte is finished! (10011011) 8 bits.

4. Convert bytes to numbers

To make sure our byte is correct were gonna calculate the byte to numbers.

1 - 128

0

0

1 - 16

1 - 8

0

1 - 2
1 - 1
-----+
155!

I hope this tutorial is kind of clear to you.
I would like to see your feedback on my first tutorial!

Greetings, ShadowReaper

Credits: Faith Connors for giving tips to make the post more detailed.

Facundo94 - November 17, 2017, 7:59 am

Great tutorial Shadow! Hope to see more like em

imscheurkie - November 17, 2017, 8:01 am

Awesome tutorial that's both concise and clear: the best kind!

Faith Connors - November 17, 2017, 9:01 am

the randomness is still not explained right. Theyre powers of two. why not mention that to make it easier to remember

ShadowReaperZ - November 17, 2017, 10:20 am

(Nov 17, 2017 10:01 AM)Faith Connors Wrote: [\[CITE\]](#) the randomness is still not explained right. Theyre powers of two. why not mention that to make it easier to rememberThank you for your feedback/reply.

I've added that the system I am talking about is the 8 bit system.
If you didn't mean this please leave a reply :)

Greetings,

Faith Connors - November 17, 2017, 10:44 am

(Nov 17, 2017 11:20 AM)ShadowReaperZ Wrote: [\[CITE\]](#) Thank you for your feedback/reply.

I've added that the system I am talking about is the 8 bit system.
If you didn't mean this please leave a reply :)

Greetings,
mathematically the numbers you gave are powers of two. this is how normal people understand number sequences. its an exponential sequence where the base of the power is two.
it doesnt matter if its 8 bit, 16 bit, 32 bit, hell fuckin 23 bit for all i care. the numbers will always be a power of two because binary means there are only two possible states

Sheeple - November 17, 2017, 12:19 pm

It's just two's complements. <https://www.cs.cornell.edu/~tomf/notes/cps104/twoscomp.html>
This "tutorial" also doesn't cover negative numbers. And if you're going to use two's complements, any number beginning with a 1 is negative.

ShadowReaperZ - November 17, 2017, 12:25 pm

(Nov 17, 2017 11:44 AM)Faith Connors Wrote: [\[CITE\]](#) mathematically the numbers you gave are powers of two. this is how normal people understand number sequences. its an exponential sequence where the base of the power is two.

it doesnt matter if its 8 bit, 16 bit, 32 bit, hell fuckin 23 bit for all i care. the numbers will always be a power of two because binary means there are only two possible states

Thank you for your quick reply,

I've added some more details.

Also I gave you credits at the bottom of the post.

Greetings,

(Nov 17, 2017 08:59 AM)Facundo94 Wrote: [\[CITE\]](#) Great tutorial Shadow! Hope to see more like em
Thank you, glad you like it.

(Nov 17, 2017 09:01 AM)imscheurkie Wrote: [\[CITE\]](#) Awesome tutorial that's both concise and clear:
the best kind!

Thank you! I am glad you like it

rektsanchez - November 17, 2017, 8:01 pm

Although i very much appreciate your effort to make this tutorial, as a newbie I have to say it is quite confusing. It seems like not all the steps are explained and some of the text blocks are quite hard to follow.