Computer Systems

CS107

Cynthia Lee

Today's Topics

LAST TIME:

- Wrap-up of C programming topics:
  - Function pointers, callbacks

THIS TIME:

- Number representation
  - Integer representation
  - Signed numbers with two's complement

NEXT TIME:

- Friday is last day of topics that will be included on next week's mditerm
  - Reasoning about special conditions with signed and unsigned
    - Overflow and underflow conditions
    - Comparison operators (< >) with signed and unsigned
  - Bytes, bits, bitwise operators

Binary


Solo hay 10 tipos
de personas en el mundo:
los que entienden binario
y los que no

Bits and Bytes

THE BUILDING BLOCKS OF EVERYTHING IN THE COMPUTER

Bits and Bytes: essential facts

- "Bit" is a **b**inary dig**it**, 0 or 1
- "Byte" is 8 bits (one `char`)
- Our system is "byte-addressable," meaning each address refers to storage space for 1 byte

- The `char, short, int, long` family of types:
  - `char` is 1 byte = 8 bits
    - 2^8 = 256 possible `char` values
  - `short` is 2 bytes = 16 bits
    - 2^{16} = 65,536 possible `short` values
  - `int` is 4 bytes = 32 bits
    - 2^{32} = 4,294,967,296 possible `int` values (~4 billion)
  - `long` is 8 bytes = 64 bits
    - 2^{64} = 18,446,744,073,709,551,616 possible `long` values (~18 quintillion)

Bits as *Unsigned* Base 2 Numbers

1000 100 10 1

1,234

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

$$
\begin{array}{r}
1000 \\
+\ 200 \\
+\ 30 \\
+\ 4 \\
\hline
1{,}234
\end{array}
$$

2^4 2^3 2^2 2^1 2^0

16 8 4 2 1

0 1 0 1 0

$$
\begin{array}{r}
8 \\
+\ 2 \\
\hline
10
\end{array}
$$

Self-test: Integer representation in binary

What is the unsigned 4-bit binary representation of 14?

a) 1111

b) 1110

c) 1010

d) Other

1 + 2 + 4 + 8 = 15

2 + 4 + 8 = 14

Self-test: Integer representation in binary

What is the base-10 equivalent of the unsigned 4-bit binary number 1010?

8 4 2 1

a) 20

b) 101

c) 10

d) 5

e) Other

Hexadecimal (base 16)

Base 10	Base 2 (4-bit)	Base 16
0	0000	0
1	0001	1
2	0010	2
3	0011	3
4	0100	4
5	0101	5
6	0110	6
7	0111	7
8	1000	8
9	1001	9
10	1010	A
11	1011	B
12	1100	C
13	1101	D
14	1110	E
15	1111	F

Self-test: Integer representation in hexadecimal

What is the unsigned binary equivalent of the unsigned hexadecimal number 0x2BEEF1?

2 B E E F 1

0010 1011 1110 1110 1111 0001

char [-128, 127]

unsigned char [0, 255]

[0, 256)

sign bit

0000 0000

sign-magnitude

1000 0000

-0

+0

↑ rejected by C!

Bits as *Signed* Base 2 Numbers

Self-test: Two's complement

What is the base-10 equivalent of the signed (two's complement) 4-bit binary number 1010?

a) -10
b) 10
c) 11
d) -11
e) 5
f) -5
g) 6
h) -6
i) Other

$-6 + 3 = -3$

$0101 + 1 = 0110$

$1010_2 = -6_{10}$

$+0011_2 = 3_{10}$

$1101 = -3_{10}$ ✓

$0010 + 1 = 0011$

$1 \mid 1101_2 = -3$

$+ \; 1101_2 = -3$

$1010 \qquad -6$

Inbox (15) - cynthia.l × | Stanford Center for P × | integer - What is the ×

stackoverflow.com/questions/94591/what-is-the-maximum-value-for-a-int32

What is the maximum value for a int32?

606

I can never remember that number. I need a memory rule.

integer

share improve this question

92

edited May 28 '14 at 14:09
Ben Hoffstein
49.5k ● 5 ● 66 ● 101

asked Sep 18 '08 at 17:18
Flinkman
5,181 ● 4 ● 18 ● 48

107 Why would you need the exact number? I remember "(2^31)-1" or "+/- 2 billion" and that's good enough for everything I ever needed. – Joachim Sauer Mar 3 '09 at 11:21

27 unsigned: 2^{32}-1 = 4·1024^3-1; signed: -2^{31} .. +2^{31}-1, because the sign-bit is the highest bit. Just learn 2^0=1 to 2^{10}=1024 and combine. 1024=1k, 1024^2=1M, 1024^3=1G – comonad Mar 28 '11 at 20:01

6 I generally remember that every 3 bits is about a decimal digit. This gets me to the right order of magnitude: 32 bits is 10 digits. – Barmar Oct 2 '13 at 15:11

30 Answers

active oldest votes

2397

It's 2,147,483,647. Easiest way to memorize it is via a tattoo.

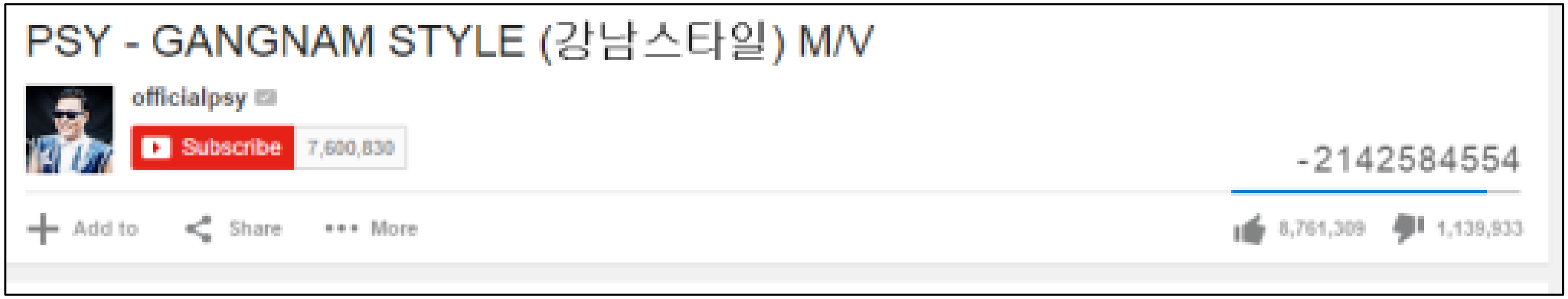
share improve this answer

edited Oct 20 '14 at 16:30
Allbite
1,415 ● 1 ● 13 ● 15

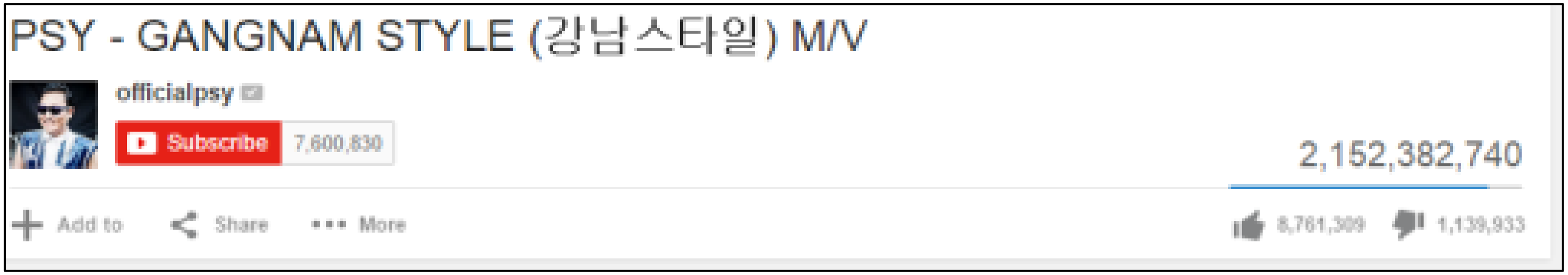
answered Sep 18 '08 at 17:20
Ben Hoffstein
49.5k ● 5 ● 66 ● 101

Overflow in two's complement

- In two's complement, when you exceed the maximum value of int (2,147,483,647), you "wrap around" to negative numbers:



- Here is the link after Google upgraded to 64-bit integers:



Signed integers with two's complement representation

Signed integers with two's complement

Goal: write 5 in 8-bit two's complement

4 2 1

$5_{10} = 101_2$

Steps to write a positive (or zero) number in two's complement:

1. Write the number in usual unsigned binary representation
2. Make sure that the number will "fit" in the number of bits you have
   - › For positive numbers, there needs to be at least one zero in the most significant (leftmost) bit
   - › 00000101 (no problem for 5 in 8 bits)
3. Done!
   - › Answer: 00000101

Signed integers with two's complement

Goal: write -5 in 8-bit two's complement

Steps to write a negative number in two's complement:

1. Write the *absolute value* of the number in usual unsigned binary representation
2. Make sure that the number will "fit" in the number of bits you have
   - Since we are writing the absolute value, a positive number, there needs to be at least one zero in the most significant (leftmost) bit*
   - 00000101 (no problem for 5 in 8 bits)
3. "Flip" each bit (0 → 1, 1 → 0)
   - 00000101 → 11111010
4. Add one
   - 11111010 → 11111011
5. Done!
   - Answer: 11111011

* There is one negative number whose positive number won't "fit"—more on this Friday

(0) Observe that # is negative → $11110100_2 = -12_{10}$

(1) Flip the sign

flip bits 0000 1011

$+$ 1

64 32 16 8 4 2 1

+1 : 0000 1100 = 12_{10}

$8 + 4 = 12$