

Recitation 3 : Datalab and Data Representations

tar xvf filename : untar command

integers - biasing

multiply : left shift by k to multiply by 2^k

divide : right shift by k to divide by 2^k for positive numbers

for negative numbers, shifting rounds towards $-\infty$, but we want to round to 0
→ biasing when negative add $2^k - 1$ to the number before we shift

1010 : -6

divide by 4, arithmetic shift by 2 (for unsigned int we use logical shift)

1110 : -2 (but we want -1)

1010 : -6

add $2^2 - 1 = 3$: 0011

1101

arithmetic shift by 2

1111 : -1

adding bias essentially preserve extra bits rather than truncate it through carry bits

Big Endian : first byte (lowest address) is the most significant
this is how we talk about binary numbers

ARM

Little Endian : first byte (lowest address) is the least significant

intel x86

0x 12345678

little : 78563412

16进制 1位 = 2进制 4位

big : 12345678

Floating Point - IEEE standard

single precision: 32 bits

s exp frac
1 8 23

double precision: 64 bits

s exp frac

1 11 52

extended precision: 80 bits (intel only)

s exp frac

1 15 63 or 64

~~(-1)^s * 2^{e-127} * (1 + f * 2⁻²³)~~