CS61 Data Representation 2

Objects

```
int local = 1
int* pointer = &local
```

- static: length of program
- automatic: allocates/frees automatically
- dynamic: allocates/frees manually

```
allocated_char = (char*) malloc(sizeof(char));
*allocated_char = 68;
```

Memory Layout

code: instructions, static lifetime

data: modifiable, static

heap: ↓ , dynamic

stack: ↓ , automatic

Data Layout

sizeof(T) = # bytes of representation type T

sizeof(x) = # bytes of object x

↑ type size_T (unsigned integer 0 to $2^{64}-1$)

Unsigned int representation

byte stored in char

☆ little-endian: least → most significant

↑ big-endian: most → least significant

x86-64

↑ internet (IP and TCP)

Signed representation

negative #s: two's complement (+/- signed uses same instructions as +/- unsigned)

to obtain $-X$: flip all bits in X ($\sim X$) and add 1

$X + (-X) = X + (\sim X + 1) = 111111\ldots1 + 1 = 0$

+ if most sig digit is 1, # is negative, $-(\sim X + 1)$
  - ↳ aka sign bit

B-bit integers: most sig bit is 2^{B-1} in unsigned, -2^{B-1} in signed

*addition, subtraction, multiplication same for signed and unsigned two's complement

division is not.

C signed arithmatic:

1. more - #s than + #s
2. arithmatic overflow on signed integers is undefined