CS 208

29 March 2023

Wednesday
bit ("binary digit")

byte (8 bits)

Fred Brooks

2 possible contents: 0 or 1

2 bits: \(2^2 = 4\) possible bit patterns

8 bits: \(2^8 = 256\) possible bit patterns
Ways of writing integers

**Base ten** ("decimal")

\[ 973 = 9 \times 10^2 + 7 \times 10^1 + 3 \times 10^0 \]

**Base two** ("Binary")

\[ 10110_{\text{two}} = 1 \times 2^4 + 0 \times 2^3 + 1 \times 2^2 + 1 \times 2^1 + 0 \times 2^0 = 22_{\text{ten}} \]

\[ 110101_{\text{two}} = ?_{\text{ten}} = 1 \times 32 + 1 \times 16 + 1 \times 4 + 1 \times 1 = 53_{\text{ten}} \]
Base sixteen "hexadecimal"

Symbols: $0123456789ABCDEF$

\[ 256 \times 16^2 + 16 \times 16 + 7 \]

\[ A \times 256 + B \times 16 + 7 \]

\[ = 10 \times 256 + 11 \times 16 + 7 \]

\[ = 2560 + 176 + 7 \]

\[ = 2743 \text{ (in ten)} \]
Why bother w/ hexadecimal?

Binary $\leftarrow$ we care

Decimal $\leftrightarrow$ Binary **hard**

Hexadecimal $\leftrightarrow$ Binary **easy**
$0_{x}\text{AB7} = \text{0b} 1010 1011 0111$

know this.
Octal - 01234567

\[ \begin{align*}
\text{Octal} & \quad \rightarrow \\
0325 & \quad \rightarrow \text{octal} \\
8^2 & \quad 8^1 & \quad 8^0 \\
64 & \quad 8 & \quad 1 \\
= 3\times64 + 2\times8 + 5 \\
= 192 + 16 + 5 & \quad = 213_{\text{ten}} \\
\end{align*} \]
Remember:

```
hexdump -C filename
```

```
man ascii
```

ASCII → conventional mapping character → integer
char *s = "emu";

8 bytes \{ 300008 \}

4 bytes

"pointer to"

x86-64
char s[10];  \[ \text{(later: \texttt{const})} \]  

1. Says "s is a char*"
2. Sets aside 10 bytes to hold the chars

char *t;

1. Says "t is a char*"
2. Sets aside 8 bytes to hold t.
2D arrays in C

char grid[10][16];

Concept
rows

columns
char grid[3][4]

Mem.

\[
grid[0][0] \\
grid[0][1] \\
grid[0][2] \ 2 \\
grid[1][2] \ 3 \\
grid[2][2] \\
\ldots
\]

"row-major order" (for storing a 2D matrix)
grid — pointer to the first char

grid[12] — pointer to the third row
