CS 208
Wed., 16 Jan 2023
\[ C_{int} \ k = 15; \]

In memory

\[ \overline{0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 1111} \]

byte \hspace{1cm} \text{byte}

Printing (paper, screen, whiteboard)

Base 10: 15

Base 16: F, f

Base 8: 17
Logical operations

AND
OR
NOT
XOR

T XOR F = T
T XOR T = F
F XOR T = T
F XOR F = F
Bitwise operations

\[
\begin{array}{c}
0101 \\ \\
\text{bitwise OR} \\
0110 \\
\hline
0111
\end{array}
\]

\[
\text{int } k = -4, \\
\text{int } j = 3;
\]

\[
\begin{array}{c}
\text{j. } 0000 \ldots 0011 \\
\rightarrow 111 \ldots 1111
\end{array}
\]
Bitwise AND

\[ k = 4 \quad 1111 \quad -- \quad 1111 \quad 1100 \]

\[ j = 3 \quad 0 \quad -- \quad 0000 \]

\[ k \& j \rightarrow 0 \]
Bitwise NOT

\[ \sim k \quad 0000 \quad 00 \quad 00 \quad 11 \]

\[ k \quad 1111 \quad 11 \quad 11 \quad 00 \]
"Boolean context" in C

```c
if ( )
while ( )
for ( - j ; ; )
```

Integers are "true" if non-zero, "false" if zero.
Why do we want bitwise operators?

A  0x41  01000001
a  0x61  01100001

B
b  01000010

Bitwise op to do to lower to upper
Mult. $\times$ (slow)
Division $\div$ (even slower)
Mod $\%$ (very slow)
Addition $+$ (fast)
Sub. $-$ (fast)

$(8 \sim 1) \rightarrow$ (super fast)
Bitwise ops in construction of network packets
Character encoding

Codepoint: an integer that represents a character.

Unicode: international agreement which codepoints go w/which characters.

$U+0041 \rightarrow A$

$U+03B1 \rightarrow \alpha$

(Greek alpha)
A scheme for storing code points as sequences of bytes
UTF-16 LE
always 2 bytes

α
U+03B1

memory

\[ \text{B1 = 1011 0001} \]
\[ \text{B2 = 0000 0011} \]
UTF-16 BE  
\[ \alpha \]
always 2 bytes

\[ \text{U+03B1} \]
UTF-16 disadvantages

- If you're programming or writing in Latin alphabet, these take up two bytes per character, a lot.

- There aren't enough codepoints that fit in two bytes. ($2^{16}$ isn't enough)
UTF-8

Ken Thompson
- UNIX (1969)
- Turing Award (1983)
- OS: Plan 9
- "Reflections on Trusting Trust"
- UTF-8

address UTF-16 disadvantages
$\text{UTF-8 } \times \text{ U+03B1}$

2 ÷ 4 × 3 row of UTF-8 chart

$1100\ldots10\ldots$