Applications of digital representation

William Hendrix

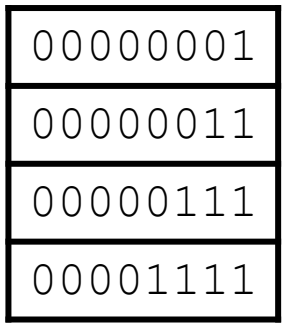
Lecture 22

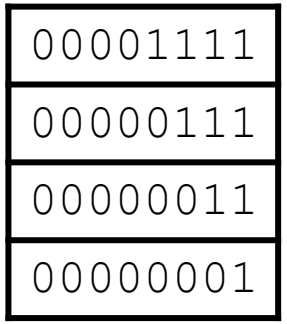
Today

- Endianness
- Bitwise xor application
- Minilab

Endianness

- How multi-byte characters are stored internally
  - Two ways
- Big Endian
  - Bytes are stored most to least significant
  - **Example:** `0x0103070f`
- Little Endian
  - Bytes are store least to most significant
  - All Intel architectures are Little Endian
  - **Example:** `0x0103070f`
  - Minor advantage: uses first byte to cast to char
    - Big Endian needs to find last byte (4^{th}/8^{th})
- Primary effects
  - Binary data cannot be exchanged between Big and Little Endian machines
  - Changes results when casting `int*` to `char*`





Bitwise xor application

- Xor truth table:

^	0	1
0	0	1
1	1	0

- `any ^ 0 = any` (no effect)
- `any ^ 1 = ~any`
  - Ones "toggle" bits
- **Application:** simple password encryption
  - Since ones toggle bits, repeating will undo changes

S	e	c	r	e	t		m	e	s	s	a	g	e	\0
k	e	y	k	e	y	k	e	y	k	e	y	k	e	y
8	\0	#26	#25	\0	#3	K	#8	#28	#24	#22	#24	#12	\0	y

  - *Trivia*: all strings in Morris worm code were XOR'ed by `0x81`
    - By setting the top bit, they no longer appeared to be letters

Minilab

- The local community theater venue has an 8 x 8 grid of seats for watching plays. Write a `Theater` class whose only field is `char seats[8]` that can reserve individual seats and confirm or cancel reservations.
  - Ensure when reserving that the seat is not already reserved
  - Don't worry about names or group reservations
  - Optional: write a print function

```
---***--
--***--*
*-*-*-**
***-*--*
-******-*
**-*****
-*******
********
```

Tonight

- **Lab 5** due Monday