Lecture Participation Poll #7

Log onto pollev.com/cse374
Or
Text CSE374 to 22333

Lecture 8: C Basics

CSE 374: Intermediate Programming Concepts and Tools
Administrivia

- HW1 deadline extended while the autograders scripts get figured out
- HW2 posted due next week

THANK YOU FOR ALL YOUR PATIENCE!
Review: Binary, Bits and Bytes

- **binary**
  - A base-2 system of representing numbers using only 1s and 0s
  - vs decimal, base 10, which has 9 symbols

- **bit**
  - The smallest unit of computer memory represented as a single binary value either 0 or 1

---

<table>
<thead>
<tr>
<th>Decimal</th>
<th>Decimal Break Down</th>
<th>Binary</th>
<th>Binary Break Down</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>((0 * 10^0))</td>
<td>0</td>
<td>((0 * 2^0))</td>
</tr>
<tr>
<td>1</td>
<td>((1 * 10^0))</td>
<td>1</td>
<td>((1 * 2^0))</td>
</tr>
<tr>
<td>10</td>
<td>((1 * 10^1) + (0 * 10^0))</td>
<td>1010</td>
<td>((1 * 2^3) + (0 * 2^2) + (1 * 2^1) + (0 * 2^0))</td>
</tr>
<tr>
<td>12</td>
<td>((1 * 10^1) + (2 * 10^0))</td>
<td>1100</td>
<td>((1 * 2^3) + (1 * 2^2) + (0 * 2^1) + (0 * 2^0))</td>
</tr>
<tr>
<td>127</td>
<td>((1 * 10^2) + (1 * 10^1) + (2 * 10^0))</td>
<td>01111111</td>
<td>((0 * 2^7) + (1 * 2^6) + (1 * 2^5) + \ldots + (1 * 2^0))</td>
</tr>
</tbody>
</table>

**byte**
The most commonly referred to unit of memory, a grouping of 8 bits
Can represent 265 different numbers (28)
1 Kilobyte = 1 thousand bytes (kb)
1 Megabyte = 1 million bytes (mb)
1 Gigabyte = 1 billion bytes (gb)
Memory Architecture

- **CPU Register**: The brain of the computer!
  - **Typical Size**: 32 bits
  - **Time**: ≈free

- **L1 Cache**: Extra memory to make accessing it faster
  - **Typical Size**: 128KB
  - **Time**: 0.5 ns

- **L2 Cache**: Extra memory to make accessing it faster
  - **Typical Size**: 2MB
  - **Time**: 7 ns

- **RAM**: Working memory, what your programs need
  - **Typical Size**: 8GB
  - **Time**: 100 ns

- **Disk**: Large, longtime storage
  - **Typical Size**: 1 TB
  - **Time**: 8,000,000 ns
RAM (Random-Access Memory)

- RAM is where data gets stored for the programs you run. Think of it as the main memory storage location for your programs.

- RAM goes by a ton of different names: memory, main memory, RAM are all names for this same thing.
RAM can be represented as a huge array

RAM:
- addresses, storing stuff at specific locations
- random access

Arrays
- indices, storing stuff at specific locations
- random access

If you’re interested in deeper than this: [https://www.youtube.com/watch?v=fpnE6UAfbtU](https://www.youtube.com/watch?v=fpnE6UAfbtU) or take some EE classes?
Working memory.

Address space: list of bytes addressed in order

- Programs are said to have access to this $2^{64}$ byte space
  - ‘64 bit’ system refers to needing 64 bits to index the space
  - But really don’t - many other things are also using this space
- Location in array is the ‘address’ of a byte
- Programs keep track of addresses of each of their pieces of memory
- Accessing unused address causes a ‘segmentation fault’
The Stack

- An area of local memory set aside to hold local variables
- Functions like the stack data structure – first in first out
- When we call a function it *allocates* memory on the stack for all local variables
  - Size of memory depends on datatype
- When the function returns the memory for the local variables is *deallocated*
- Java has been doing something similar in the background for you all along - garbage collector
Blinky and Pointer Fun!