Instructor: Justin Hsia

Teaching Assistants:
Adithi Raghavan
Aman Mohammed
Connie Chen
Eyoel Gebre
Jiawei Huang
Malak Zaki
Naama Amiel
Nathan Khuat
Nikolas McNamee
Pedro Amarante
Will Robertson

http://xkcd.com/953/
Relevant Course Information

❖ Upcoming deadlines
  ▪ Pre-Course Survey and HW0 due tonight
  ▪ HW1 due Monday (1/8)
  ▪ Lab 0 due Monday (1/8)
    • This lab is *exploratory* and looks like a HW; the other labs will look a lot different

❖ Ed Discussion etiquette
  ▪ For anything that doesn’t involve sensitive information or a solution, post publicly (you can post anonymously!)
  ▪ If you feel like you question has been sufficiently answered, make sure that a response has a checkmark
EPA

❖ Encourage class-wide learning!

❖ Effort
  ▪ Attending support hours, completing all assignments
  ▪ Keeping up with Ed Discussion activity

❖ Participation
  ▪ Making the class more interactive by asking questions in lecture, section, support hours, and on Ed Discussion

❖ Altruism
  ▪ Helping others in section, support hours, and on Ed Discussion
Memory & Data I
Lesson Summary (1/2)

❖ Memory is a long, *byte-addressed* array
  - Word size bounds the size of the *address space* and memory
  - Address of a chunk of memory given by the address of the lowest byte in chunk

❖ Endianness determines memory storage order for multi-byte data
  - Least significant byte in lowest (little-endian) or highest (big-endian) address of memory chunk

![Memory diagram](image)
Lesson Summary (2/2)

❖ Programming Data

- Variable declaration allocates space for data type size
- Assignment results in value being put in memory location

<table>
<thead>
<tr>
<th>C Data Type</th>
<th>x86-64 Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>char</td>
<td>1B</td>
</tr>
<tr>
<td>short</td>
<td>2B</td>
</tr>
<tr>
<td>int</td>
<td>4B</td>
</tr>
<tr>
<td>long</td>
<td>8B</td>
</tr>
<tr>
<td>long long</td>
<td>8B</td>
</tr>
<tr>
<td>float</td>
<td>4B</td>
</tr>
<tr>
<td>double</td>
<td>8B</td>
</tr>
<tr>
<td>long double</td>
<td>16B</td>
</tr>
</tbody>
</table>
Lesson Q&A

❖ Learning Objectives:
  ▪ (Define the concept of pointers and) their significance in computer memory organization.
  ▪ (Design code that can correctly) interpret and manipulate multi-byte data in both little-endian and big-endian byte orderings.

❖ What lingering questions do you have from the lesson?
  ▪ Chat with your neighbors about the lesson for a few minutes to come up with questions
Memory & Data I — Practice
Polling Questions (1/2)

❖ By looking at the bits stored in memory, I can tell what a particular 4 bytes is being used to represent.
   A. True   B. False

❖ We can fetch a piece of data from memory as long as we have its address.
   A. True   B. False

❖ Which of the following bytes have a most-significant bit (MSB) of 1?
   A. 0x63   B. 0x90   C. 0xCA   D. 0xF
Polling Questions (2/2)

❖ We store the value 0x 01 02 03 04 as a **word** at address **0x100** in a big-endian, 64-bit machine

❖ What is the **byte of data** stored at address **0x104**?

A. 0x04
B. 0x40
C. 0x01
D. 0x10
E. We’re lost...
Homework Setup

❖ Assume that a snippet of memory is shown below (in hex), starting with the byte at address 0x08 on a little-endian machine:

```
addr:   0x08 0x09 0x0A 0x0B 0x0C 0x0D 0x0E 0x0F
data:   | A2 | D0 | 4F | C4 | A0 | 0C | F7 | 27 |
```

❖ What is the value of the int stored at address 0x0C?
Modern System Details

- Current x86-64 systems use **64-bit (8-byte) words** ("64-bit machines")
  - Potential address space: $2^{64}$ addresses
    - $2^{64}$ bytes $\approx 1.8 \times 10^{19}$ bytes
    - $= 18$ billion billion bytes $= 18$ EB (exabytes)
  - Actual physical address space: **48 bits**
    - This is sufficient space for now and allows for some operating system tricks
    - Example address: $0x\ 7f\ f\ c\ 3d\ d5\ 06\ 94$

- There’s a lot more to this story... stay tuned for virtual memory!
Discussion Question

❖ Discuss the following question(s) in groups of 3-4 students
  ▪ I will call on a few groups afterwards so please be prepared to share out
  ▪ Be respectful of others’ opinions and experiences

❖ Over time, computers have grown in word size:

<table>
<thead>
<tr>
<th>Word size</th>
<th>Instruction Set Architecture</th>
<th>First? Intel CPU</th>
<th>Year Introduced</th>
</tr>
</thead>
<tbody>
<tr>
<td>8-bit</td>
<td>??? (Poor &amp; Pyle)</td>
<td>Intel 8008</td>
<td>1972</td>
</tr>
<tr>
<td>16-bit</td>
<td>x86</td>
<td>Intel 8086</td>
<td>1978</td>
</tr>
<tr>
<td>32-bit</td>
<td>IA-32</td>
<td>Intel 386</td>
<td>1985</td>
</tr>
<tr>
<td>64-bit</td>
<td>IA-64</td>
<td>Itanium (Merced)</td>
<td>2001</td>
</tr>
<tr>
<td>64-bit</td>
<td>x86-64</td>
<td>Xeon (Nocona)</td>
<td>2004</td>
</tr>
</tbody>
</table>

▪ What do you think were some of the causes, advantages, and disadvantages of this trend?
Group Work Time

❖ During this time, you are encouraged to work on the following:
   1) If desired, continue your discussion
   2) Work on the homework problems
   3) Work on the lab (if applicable)

❖ Resources:
   ▪ You can revisit the lesson material
   ▪ Work together in groups and help each other out
   ▪ Course staff will circle around to provide support