Bits, Bytes, and Integers – Part 1

15-213/18-213/15-513: Introduction to Computer Systems
2nd Lecture, May 22, 2019

Instructors:
Sol Boucher
But first… an overview of course topic progression and labs
Programs and Data

Topics
- Bit operations, arithmetic, assembly language programs
- Representation of C control and data structures
- Includes aspects of architecture and compilers

Assignments
- L1 (devicelab): Manipulating bits, characters, and strings
- L2 (bomblab): Defusing a binary bomb
- L3 (attacklab): The basics of code injection attacks

Note: new first lab vs. past terms, current 513!
The Memory Hierarchy

Topics

- Memory technology, memory hierarchy, caches, locality
- Includes aspects of architecture and OS

Assignments

  - Learn how to exploit locality in your programs.
Memory Allocation

Topics
- Dynamic storage allocation, virtual memory, address translation
- Includes aspects of architecture and OS

Assignments
- L5 (malloclab): Writing your own malloc package
  - Get a real feel for systems-level programming

Note: different topic/lab order vs. past terms!
Exceptional Control Flow

Topics
- Hardware exceptions, processes, process control, Unix signals, nonlocal jumps
- Includes aspects of compilers, OS, and architecture

Assignments
- L6 (tshlab): Writing your own Unix shell.
  - A first introduction to concurrency
Networking, and Concurrency

Topics

- High level and low-level I/O, network programming
- Internet services, Web servers
- concurrency, concurrent server design, threads
- I/O multiplexing with select
- Includes aspects of networking, OS, and architecture

Assignments

- L7 (proxylab): Writing your own Web proxy
  - Learn network programming and more about concurrency and synchronization.
Lab Rationale

Each lab has a well-defined goal such as solving a puzzle or winning a contest.

Doing the lab should result in new skills and concepts.

We try to use competition in a fun and healthy way:
- Set a reasonable threshold for full credit
- Post intermediate results (anonymized) on Autolab scoreboard for glory!
Policies: Grading

- Exams (50%): midterm (20%), final (30%)

- Labs (50%): weighted according to effort

- Final grades based on a straight scale (90/80/70/60) with a small amount of curving
  - Only upward
Doing the Lab

- [ ] https://autolab.andrew.cmu.edu/courses/15213-m18
  - Hosts a writeup with instructions for downloading and completing the lab
  - Access will be granted on Friday when the first lab is released

If you have questions

- [ ] Piazza
- [ ] Office hours...
Office Hours

Prof. Railing
- Immediately after lectures, i.e. class meetings except:
  - Bootcamps
  - Recitations
  - Exam reviews

Prof. Boucher
- Wednesdays at 3 PM in GHC–8115
- Fridays at 3 PM in GHC–9115

TAs
- Primary time TBA

Availability may vary week to week
- Check Office Hours page on website
Lecture Schedule Preview

This week: *Everything* you need for devicelab!

- Yesterday: Course Overview (Prof. Railing)
- **Today:** Bits, Bytes, Integers (Prof. Boucher)
- Tomorrow: *More* Bits, Bytes, Integers (Prof. Boucher)
- Friday: Floating Point (Prof. Railing)

Next week

- Tuesday: Linux/Git bootcamp (Tas)
- ...
Waitlist questions

- 15-213: Amy Weis alweis@andrew.cmu.edu
- 18-213: Zara Collier zcollier@andrew.cmu.edu
- 15-513: Amy Weis alweis@andrew.cmu.edu

Please don’t contact the instructors with waitlist questions.
Today/tomorrow: Bits, Bytes, Integers

Representing information as bits

Integers
  - Representation: unsigned and signed
  - Bit-level manipulations
  - Conversion, casting
  - Expanding, truncating
  - Addition, negation, multiplication, shifting
  - Summary

Representations in memory, pointers, strings
Everything is bits

Each bit is 0 or 1

By encoding/interpreting sets of bits in various ways
- Computers determine what to do (instructions)
- ... and represent and manipulate numbers, sets, strings, etc...

Why bits? Electronic Implementation
- Easy to store with bistable elements
- Reliably transmitted on noisy and inaccurate wires

![Diagram showing voltage levels and binary states](image-url)
For example, can count in binary

**Base 2 Number Representation**

- Represent $15213_{10}$ as $11101101101101_2$
- Represent $1.20_{10}$ as $1.0011001100110011[0011]..._2$
- Represent $1.5213 \times 10^4$ as $1.1101101101101_2 \times 2^{13}$
Encoding Byte Values

**Byte = 8 bits**

- Binary: 00000000₂ to 11111111₂
- Decimal: 0₁₀ to 255₁₀
- Hexadecimal: 00₁₆ to FF₁₆
  - Base 16 number representation
  - Use characters ‘0’ to ‘9’ and ‘A’ to ‘F’
  - Write FA1D37B₁₆ in C as
    - 0xFA1D37B
    - 0xfa1d37b

### Activity: binary, hexadecimal, two’s complement

15213: 0011 1011 0110 1101

- 3
- B
- 6
- D
## Example Data Representations

<table>
<thead>
<tr>
<th>C Data Type</th>
<th>Typical 32-bit</th>
<th>Typical 64-bit</th>
<th>x86-64</th>
</tr>
</thead>
<tbody>
<tr>
<td>char</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>short</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>int</td>
<td>4</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>long</td>
<td>4</td>
<td>8</td>
<td>8</td>
</tr>
<tr>
<td>float</td>
<td>4</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>double</td>
<td>8</td>
<td>8</td>
<td>8</td>
</tr>
<tr>
<td>pointer</td>
<td>4</td>
<td>8</td>
<td>8</td>
</tr>
</tbody>
</table>

Preview: negative numbers and fixed widths