15-213
“The Class That Gives CMU Its Zip!”

Introduction to Computer Systems

Taerim Kim
January 23, 2012

Topics:

- Welcome
- Integers
- Datalab

Welcome to recitation

Where am I?

- 15-213 recitation, section G
- Somewhere in Wean...

Who is this fool standing in front of me?

- Taerim Kim
- SCS, class of 2012
- Fourth semester on 15-213 staff (or something like that)

What are we supposed to be doing here?

- Reinforce important lecture materials
- Jump start on labs
- Review for exams

Getting help

Recitations

- Come with questions
- Ask questions
- Pay attention

Office hours

- Monday through Thursday
- 6:00pm to 8:00pm
- WEH 5207

Piazza

- Replaces the staff mailing list
- Students post questions—answered by fellow students or staff members

Posting on Piazza

Answers to these questions should be, “Yes”

- Have I read the textbook?
- Have I read the writeup?
- Have I read the FAQs?
- Have I read relevant *man* pages?
- Have I spent a nontrivial amount of time pondering the problem myself?

Answers to these questions should be, “No”

- Will I have to post lots of code?
- Will I give anything away to my peers?

Review of integers

Two types

- Signed
- Unsigned

Consider a *k*-bit unsigned integer

- Bit *n* has value ____
- Minimum value is ____, and maximum value is ____

Consider a *k*-bit signed integer

- Bit *n* has value ____, except for most significant bit, which has value ____
- Minimum value is ____, and maximum value is ____

Shifting

Two directions

- Left: `x << k`
- Right: `x >> k`

Two types

- Arithmetic
- Logical

Relationship with signed/unsigned

- Which type of shift is used with each type of integer?
- Why?

Thinking about shifting

Left shift

- Similar to which arithmetic operation?
- Completely identical?

Right shift

- Similar to which arithmetic operation?
- Completely identical?

The benefit of shifting

- Often faster than performing an actual arithmetic operation
- Good compilers will replace when appropriate on programmer's behalf

Nifty two's complement factoids

Negation

- Does `-x == ~x + 1`?
- Always?

Properties of zero

- What is the value of `x & 0`?
- What is the value of `x | 0`?
- What is the value of `x & (0 – 1)`?
- What is the value of `-0`?

Datalab

Details

- Available since Thursday, January 19
- Due Thursday, February 2
- That's two weeks! You shouldn't need all that time, but don't procrastinate.

Your job

- Solve a series of integer and floating point puzzles
- Use as few operations as possible
- Learn stuff
  - Representation of numbers in computers
  - Manipulating numbers

Starting a lab

Set up your workspace

- Download the handout
- Unpack the handout
- “Keep it secret. Keep it safe.”

Read relevant stuff

- Read the textbook
- Read the writeup
- Read the textbook
- Read the writeup

Work on the shark machines

Unix basics

Know how to remotely access lab machines

- SCP (WinSCP if you're on Winblows)
- SSH (PuTTY if you're on Winblows)

Familiarize yourself with a reasonable text editor

- Emacs (lol)
- Vim

Figure out Unix and AFS file permissions

Learn how to help yourself

- *man*
- *apropos*

A parting gift: a hint

Don't underestimate datalab

- One puzzle in particular is very difficult
- You will struggle with the limit on the number of operations

Think about this example

- Write the function `int parity(int n);`
- Returns `1` if `n` has an odd number of `1` bits; `0` otherwise
- How do we do that?

A parting gift: a hint

Don't underestimate datalab

- One puzzle in particular is very difficult
- You will struggle with the limit on the number of operations

Think about this example

- Write the function `int parity(int n);`
- Returns `1` if `n` has an odd number of `1` bits; `0` otherwise
- How do we do that?
  - Idea: XOR all of the bits together!
    - » Example: `10100111`
    - » `1 ^ 0 ^ 1 ^ 0 ^ 0 ^ 1 ^ 1 ^ 1 = 1`
  - But this uses way too many operations

A parting gift: a hint

Don't underestimate datalab

- One puzzle in particular is very difficult
- You will struggle with the limit on the number of operations

Think about this example

- Write the function `int parity(int n);`
- Returns `1` if `n` has an odd number of `1` bits; `0` otherwise
- How do we do that?
  - Idea: divide and conquer!
    - » Example: `10100111`
    - » `1010 ^ 0111 = 1101`
    - » `11 ^ 01 = 10`   `1 ^ 0 = 1`

Go away