Welcome !

COMP 273 Winter 2016

Introduction to Computer Systems

Prof: Michael Langer

Who am I ?

- grew up in Toronto

- BSc at McGill in early 1980s (Math Major, Minor CS)
- MSc in CS at U of Toronto in late 1980s
- PhD (in ECE) at McGill in early 1990s

- postdoctoral research 1995-1999
  - computer vision, at NEC Research in NJ, USA
  - human vision, at MPI in Germany

- professor here since 2000
  http://www.cim.mcgill.ca/~langer/

- research topics
  - Computational Models of Human Vision (mostly depth perception)
  - Computer Vision
  - Applications of Visual Perception in Computer Graphics

Who are you ?

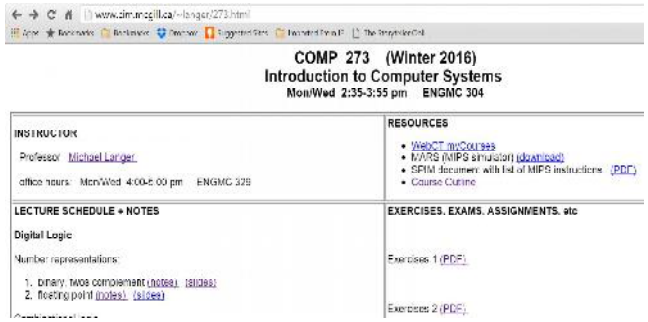
B Sci	120
B Arts	40
B Arts & Sci	10
Misc	10

U0	10	
U1	25	(returning)
U1	50	(new)
U2	50	
U3	45	

Course Public Web page

http://www.cim.mcgill.ca/~langer/273.html

Yes, lecture recordings will be made available, as will PDF lecture notes and exercises.



Overview of Course

1. Digital Logic (~ 7 lectures)

   How are numbers represented in a computer?
   How are arithmetic and logical operations performed ?
   How does data flow through a circuit ?
   How is data held in memory ?
   What is a clock ? Why is computer time discrete ?

2. MIPS Assembly language (~ 5 lectures)

   What is a "low level" programming language ?
   How are instructions represented ?
   How are data/instructions laid out in memory ?
   How are application vs. OS programs related ?
   How are function/method calls implemented? (+recursion)

3. MIPS Data Paths and Control (~3 lectures)

   What is relationship between parts 1 and 2 ?

4. Memory (~ 3 lectures)

   What types of physical memory are there, *and why* ?
   (registers, cache, RAM, disk)
   What is the relationship between the programmer's model of memory ("virtual") vs. physical memory ?

5. Input/Output (I/O) (~4 lectures)

   How do the various parts of a computer interact ?
   (disk, keyboard, monitor, mouse, USB ports)

6. Topics (~3 lectures)

Co-Requisite Courses

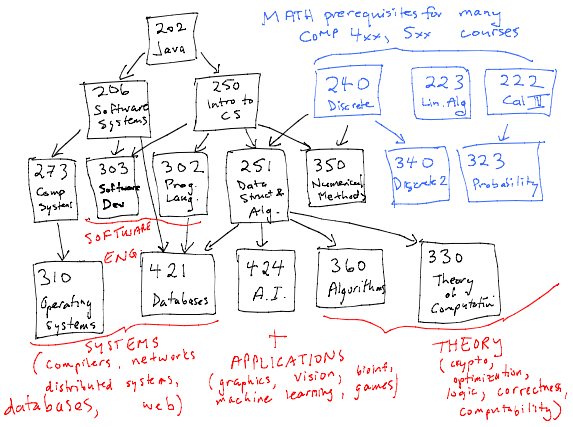
You should either have taken or be taking:

- (official) COMP 206 Introduction to Software Systems

  It will help to have some familiarity with
  - bit operations (C)
  - notion of 'system call' (C, UNIX)
  - lower level programming
    (C pointers vs arrays, malloc and free)

- (unofficial) COMP 250 Introduction to CS
  - you need to know about recursion, what is a stack?

How does this course fit into the COMP curriculum streams ?



Evaluation

- 20% for quizzes

  There will be 6 quizzes (schedule TBD)
  Each will be 15 minutes, at start of class.
  I will count your best 5 scores (5 x 4% = 20%)

- 30% assignments

  3 x 10% each (MIPS programming)

- 50% final exam

  (or 70%, namely replace all your quiz grades)

Course Workload

McGill Faculty of Science has an official policy on workload
http://www.mcgill.ca/study/2015-2016/sites/mcgill.ca.study.2015-2016/files/science_undergraduate_2015-2016_2nd_edition_0.pdf

"The credit assigned to a particular course should reflect the amount of effort it demands of you. Normally, one credit will represent three hours total work per week for one term—including a combination of lecture hours, other contact hours, such as laboratory periods, tutorials, and problem periods, as well as personal study time."

117 hours = 3 hours/credit/week x 13 weeks x 3 credits

~ 40 (lectures) + 40 (review) + 40 (assignments)

Instructor Availability

- office hours (after class: MW 4-5 pm in ENGMC 329)
- by appointment
- by email: michael.langer@mcgill.ca, langer@cim.mcgill.ca
  - you should use your mcgill address to avoid spam filtering
  - are forwarding from firstname.lastname@mail.mcgill.ca ?

COMP 273 myCourses Discussion Board

- you need to subscribe to be automatically informed
- for your postings, please:
  - choose subject line
  - respect the threads
  - new idea -> new thread please!
  - don't post public "thank you" notes (send privately)

Facebook ? (I will almost never post there.)

Plagiarism on Assignments

Plagiarism ~ presenting someone else's work as your own

Q: Where to draw the line between collaboration and plagiarism? (Different profs have different attitudes. If you are unsure, then ask.)

A: My attitude:

Public discussion is allowed e.g. MyCourses
(Anyone can listen in, including me.)

Private discussion is allowed, but sharing code or giving away main ideas is NOT.

Let's get started !

Q: How do people and computers represent numbers ?

Base 10

$$238 = 2 \cdot 10^2 + 3 \cdot 10^1 + 8 \cdot 10^0$$

$$m = \sum_{i=0}^{k-1} a_i \, 10^i$$

$$= (a_{k-1} \, a_{k-2} \, \cdots \, a_2 \, a_1 \, a_0)_{ten}$$

Base 2

$$11010 \equiv 1 \cdot 2^4 + 1 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 0 \cdot 2^0$$

$$= 16 + 8 + 2 = 26$$

$$m = \sum_{i=0}^{n-1} b_i \, 2^i$$

$$= (b_{n-1} \, b_{n-2} \, \cdots \, b_2 \, b_1 \, b_0)_{two}$$

Counting in binary

decimal	binary	binary
0	0	0000
1	1	0001
2	10	0010
3	11	0011
4	100	0100
5	101	0101
6	110	0110
7	111	0111
8	1000	1000
⋮	⋮	

To convert from binary to decimal, you need to know the powers of 2.

n	2^n
0	1
1	2
2	4
3	8
4	16
5	32
6	64
7	128
8	256
9	512
10	1024
⋮	⋮

memorize

How can we convert **m** from decimal to binary?

Idea 1:

Find the biggest power of two less than or equal to m, and subtract it from m.

Repeat until done.

(Requires memorizing powers of 2.)

How can we convert **m** from decimal to binary?

Idea 2: Consider familiar idea from base 10:

$$238 = 230 + 8$$

$$m = (m/10) \times 10 + m \% 10$$

integer division

Same idea works in base 2.

$$m = (m/2) \times 2 + m \% 2$$

Example:

$$m = (1001\,1)_{two}$$

$$m/2 = 1001$$

$$m \% 2 = 1$$

$$m = \sum_{i=0}^{n-1} b_i 2^i$$

$$= (b_{n-1} b_{n-2} \cdots b_2 b_1 b_0)_{two}$$

$$m/2 = \sum_{i=1}^{n-1} b_i 2^{i-1}$$

$$= (b_{n-1} b_{n-2} \cdots b_2 b_1)_{two}$$

$$m \% 2 = b_0$$

Algorithm:
given m in decimal, convert it to binary.

```
i ← 0
while m > 0 {
    b_i ← m % 2
    m ← m / 2
    i ← i + 1
}
```

Example $m = 241 = (1111\,0001)_{two}$

m	b_i
241	
= 120 × 2 +	1
60	0
30	0
15	0
7	1
3	1
1	1
0	1

m	b_i
$(1111\,0001)_{two}$	
1111000	1
111100	0
11110	0
1111	0
111	1
11	1
1	1
	1

Q: How to add two numbers in binary ?

Addition (base 10)

```
  1 0 1   ← carry
  2 3 4 3
+ 5 8 1 9
-------
  8 1 6 2
```

You need to memorize single digit sums to do this.

Addition (base 2)

```
 0 0 1 1 0 1 0
 0 0 0 1 1 0 1 0   ← 26
 0 0 0 1 1 0 1 1   ← 27
 ---------------
 0 0 1 1 0 1 0 1   ← 53
```

Subtraction (base 10)

```
  2 3 4 3
- 5 8 1 9
---------
```

ASIDE: the grade school algorithm doesn't work when the bigger number is on the bottom. To take the difference using the grade school algorithm, you put the bigger number on top and take the negative of the result.

$$a - b$$

$$= a + (-b)$$

Next class we will learn how to represent *negative numbers in binary* which allows us to perform this sum.