Topic 1: Overview of Computer Organization and Systems Programming

CSE 30: Computer Organization and Systems Programming
Fall 2015

Diba Mirza
Dept. of Computer Science and Engineering
University of California, San Diego
Information about the Instructor/TAs

- Instructor: Diba Mirza (Adjunct Assistant Professor, CSE)
  - Address me as Professor Mirza or Dr. Mirza
- Office: 2124 EBU3B
- Email: dimirza@eng.ucsd.edu
- Office hours:
  - MW 10am -11am, 1:00pm – 2:00pm
  - Or by appointment
- TAs: Aravind, Bilal, Ibrahim, Jorel, Josh, Srinivas
- Tutors: Alvin, Audrey, Brianna, Christian, Di, Dylan, Gokul, Ivan, Joshua (Josh), Jules, Kimberly, Melissa, Timothy (Tim) …
- TA/tutor lab office hours will be made available on the course calendar shortly.
Goals of the course

- Understand how a computer works – why? how?
  - Look under the hood of high-level programs
  - Learn the language of the machine – Assembly!
- Learn C – why?
- Understand the limits of a computer and what affects the performance of your programs
- Learn big ideas that have shaped computing
Logistics: Course Components

https://cseweb.ucsd.edu/classes/fa15/cse30/index.html

<table>
<thead>
<tr>
<th>Component</th>
<th>Weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>HW/PA Assignments (4)</td>
<td>25% (4% + 7% x 3)</td>
</tr>
<tr>
<td>Midterm (2)*</td>
<td>40% (20% + 20%)</td>
</tr>
<tr>
<td>Final (1)</td>
<td>25%</td>
</tr>
<tr>
<td>Online Quiz (10), every Friday</td>
<td>8%</td>
</tr>
<tr>
<td>Class participation (Clickers)</td>
<td>2%</td>
</tr>
</tbody>
</table>

*Look out for Piazza poll to help with midterm logistics
In class we will use Clickers!

- Lets you vote on multiple choice questions in real time.
Lecture: Peer Instruction

- I will pose carefully designed questions. You will
  - Solo vote: Think for yourself and select answer
  - Discuss: Analyze problem in teams of two or three
    - Practice analyzing, talking about challenging concepts
    - Reach consensus
    - If you have questions, raise your hand and I will come over
  - Group vote: Everyone in group votes
    - You must all vote the same to get your point
  - Class wide discussion:
    - Led by YOU (students) – tell us what you talked about in discussion that everyone should know!
Why Peer Instruction?

- You get to make sure you are following the lecture.
- I get feedback as to what you understand.
- It’s less boring!
- Research shows it promotes more learning than standard lecture.

Take a minute to introduce yourself to your group
Familiarity with C

How familiar are you with C?

A. I know C pretty well and have a lot of programming experience
B. Reasonably well, lots of gaps
C. I don’t know what a pointer is
Course Schedule (first two weeks)

<table>
<thead>
<tr>
<th>Sun</th>
<th>Mon</th>
<th>Tue</th>
<th>Wed</th>
<th>Thu</th>
<th>Fri</th>
<th>Sat</th>
</tr>
</thead>
<tbody>
<tr>
<td>27</td>
<td>28</td>
<td>29</td>
<td>30</td>
<td>Oct 1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>9am Sec A: Lecture</td>
<td>8am Sec A: Discussion</td>
<td>Quiz: Due 10pm</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10am Diba's Office</td>
<td>9am Sec A: Lecture</td>
<td>9am Sec A: Lecture</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12pm Sec B: Lecture</td>
<td>10am Diba's Office</td>
<td>12pm Sec B: Lecture</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1pm Diba's Office</td>
<td>12pm Sec B: Lecture</td>
<td>1pm Diba's Office</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3pm Sec C: Lecture</td>
<td>1pm Diba's Office</td>
<td>3pm Sec C: Lecture</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PA1 Due: 10pm</td>
<td>2pm Sec B: Discussion</td>
<td>5pm Sec C: Discussion</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8am Sec A: Lecture</td>
<td>3pm Sec C: Lecture</td>
<td>8pm Sec C: Discussion</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10am Diba's Office</td>
<td>3pm Sec C: Lecture</td>
<td>Quiz: Due 10pm</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12pm Sec B: Lecture</td>
<td>3pm Sec C: Lecture</td>
<td>9pm Sec C: Discussion</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1pm Diba's Office</td>
<td>5pm Sec C: Discussion</td>
<td>10pm Sec C: Discussion</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3pm Sec C: Lecture</td>
<td>8pm Sec C: Discussion</td>
<td>10pm Sec C: Discussion</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
ARM based embedded platform!

Q: Have you purchased your Pi?
   A. Yes
   B. No
About PAs

- If you have purchased your Pi and are yet to get set up:
  - Go to labs next week and attend discussion-Wednesday!
- If you haven’t purchased your Pi, do so immediately.
About PAs

- Pair programming (optional)
  - Choose your partner from any of the three sections by Monday (10 am)
  - You cannot change your partner for the duration of the quarter
  - Possible to break partnerships with advance 2 day notice in WRITING, in which case you would complete subsequent PAs individually
- Agree to meet at a particular place and time
- If you don’t honor your agreements more than once, you are subject to disciplinary action
- Alternate ownership of the Pi
Course Problems...Cheating

- What is cheating?
  - Studying together in groups is encouraged
  - Turned-in work must be *completely* your own.
  - Common examples of cheating: running out of time on an assignment and then pick up output, take homework from box and copy, person asks to borrow solution “just to take a look”, copying an exam question, …
  - Both “giver” and “receiver” are equally culpable

- Cheating on PA and HW/ exams; *In most cases, F in the course.*
- Any instance of cheating will be referred to Academic Integrity Office
Grading structure and policy

- I will follow a standard grading scale, subject to discretion from my end.
- What do I need to get an A- or better?
  - >90% overall

- What do I need to get (C- or better)?
  - >70% overall AND
  - Must appear on Final exam
Logistics: References

- Required textbook (rent the e-book from bookstore for $20 for the whole quarter):
  ARM Assembly Language: Fundamentals and Techniques, 2nd edition, William Hohl

  Recommended textbook:
  The C Programming Language, Kernighan and Ritchie, 2nd edition
The Evolution of Computing

2400 BC

17th Century

1804

1822

Automated textile looms

Pascaline

Schickard’s Machine

Jacquard’s Loom

Analytical Engine

2400 BC

17th Century

1804

1822
Big Idea behind early ‘computers’

Fixed Program Model

Specific (computation) Problem

Circuit to solve it

- The ‘program’ was wired into the computing device
Next big idea... The stored program model

• Key Idea(s):
  • Computer divided into two components: Processor and Memory
  • Program and data stored in the same place: memory

Have a new problem?
• Don’t change the machine
• Change the recipe

Recipe/ Program
(Set of instructions)

Problem

Processor

Memory
(Program)

Stored Program Model
proposed by Jon Von Neumann
The Von Neumann Architecture

4 Basic Components of a Computer:

1. **Memory**: a long but finite sequence of cells (1D)
   - Each cell has a distinct address
   - Data in each cell: instruction, data or the address of another cell
2. **Control Unit**: Fetches instructions from memory and decodes them
3. **Arithmetic Logic Unit**: Does simple math operations on data
4. **Input/Output**: The connections with the outside world
The Evolution of Computing

Revolution:
1st Large Scale, General Purpose Electronic Computer

ENIAC

- More complex electronic circuits
- Solved more general problems
- Programming involved configuring external switches or feeding instructions through punched cards

WWII  The stored program model
The Evolution of Computing

Revolution: Integrated Circuit:
Many digital operations on the same material

Vacuum tubes

ENIAC

Stored Program Model

WWII

1949

Integrated Circuit

(1.6 x 11.1 mm)

Exponential Growth of Computation

Moore’s Law

1965
Technology Trends: Microprocessor Complexity

In 1965, Gordon Moore predicted that the number of transistors per chip would double every 18 months (1.5 years)

Gordon Moore
Intel Cofounder
Exponential growth in computing
Computer Technology – Dramatic Change!

- Memory
  - DRAM capacity: 2x / 2 years (since ‘96); 64x size improvement in last decade.

- Processor
  - Speed 2x / 1.5 years (since ‘85); 100X performance in last decade.

- Disk
  - Capacity: 2x / 1 year (since ‘97); 250X size in last decade.
Current State of Computing

- Computers are cheap, embedded everywhere
- Transition from how to we build computers to how to we use computers

eMerging Societal-Scale Systems

New System Architectures
New Enabled Applications
Diverse, Connected, Physical, Virtual, Fluid

Embedded Systems
MEMS BioMonitoring
Information Appliances

"Server"
"Client"
Scalable, Reliable, Secure Services
“The use of [these embedded computers] throughout society could well dwarf previous milestones in the information revolution.”
How do we handle complexity?

- Big idea: Coordination of many *levels of abstraction*
Levels of Representation

High Level Language Program (e.g., C)

Assembly Language Program (e.g., ARM)

Machine Language Program (ARM)

Compiler

Assembler

Machine Interpretation

Hardware Architecture Description (e.g., block diagrams)

Architecture Implementation

Logic Circuit Description (Circuit Schematic Diagrams)

temp = v[k];
v[k] = v[k+1];
v[k+1] = temp;

\texttt{ldr \ r0, [r2]}
\texttt{ldr \ r1, [r2, #4]}
\texttt{str \ r1, [r2]}
\texttt{str \ r0, [r2, #4]}

\begin{verbatim}
0000 1001 1100 0110 1010 1111 0101 1000
1010 1111 0101 1000 0000 1001 1100 0110
1100 0110 1010 1111 0101 1000 0000 1001
0101 1000 0000 1001 1100 0110 1010 1111
\end{verbatim}
Abstraction is good – but …

- We still need to understand the system!
- As a programmer you will be manipulating data.
- Data can be anything: numbers (integers, floating points), text, pictures, video!
- Writing efficient code involves understanding how “data” and programs are actually represented in memory
- Next class we’ll talk about the bits and bytes of data: Number representation