Inf2C - Computer Systems
Lecture 1
Course overview & the big picture

Boris Grot
School of Informatics
University of Edinburgh
Practicalities

- Lectures:
  - Tue 15:10 – 16:00 @ Appleton Tower, LT 2
  - Fri 15:10 – 16:00 @ Teviot LT

- Tutorials: weeks 3, 5, 7, 9

- Labs: Drop-in format.
  - Demonstrators available to help

- Online discussion forum: Piazza
  - [https://piazza.com/ed.ac.uk/fall2017/inf2ccs/home](https://piazza.com/ed.ac.uk/fall2017/inf2ccs/home)
  - Primary means to Q&A outside of class.

- Study resources: slides, book, lecture notes (online)

- All material are/will be on the course web-page: [http://www.inf.ed.ac.uk/teaching/courses/inf2c-cs](http://www.inf.ed.ac.uk/teaching/courses/inf2c-cs)
Lecture schedule, slides, notes

INF2C COMPUTER SYSTEMS 2017-18: SCHEDULE

The abbreviations used in the Reading column are:


<table>
<thead>
<tr>
<th>Wk</th>
<th>Date</th>
<th>Lecture Topic</th>
<th>Reading</th>
<th>Tutorial</th>
<th>Lab</th>
<th>Coursework</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Tue 19 Sep</td>
<td>Introduction: the Big Picture</td>
<td>P &amp; H 1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>(Notes)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>Fri 22 Sep</td>
<td>Data Representation 1</td>
<td>P &amp; H 3/e: 3.1-3.3, 3.6 (up to FP add)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>(Notes)</td>
<td>P &amp; H 4/e: 2.4, 3.1, 3.2, 3.5 (up to FP add)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>P &amp; H 5/e: 2.4, 3.1, 3.2, 3.5 (up to FP add)</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Schedule will drift. It’s OK.
Books

- **Required:** *Patterson & Hennessy: Computer Organization and Design, Morgan Kaufmann*
  - 5th or 4th ed recommended
  - Several copies on reserve in the library and online

  - Library has 5th and 7th ed ebook (both OK)
  - Only a few sections needed for this course

- *Kernighan and Ritchie. The C Programming Language, Prentice Hall 2nd ed*
  - Generally useful, but not mandatory for this course
Exam and Coursework

- Exam - 60%
  - In December; exact date not available yet.
  - Must achieve at least 40/100 to pass the course

- Coursework – 40%
  1. MIPS assembly programming
     - Out: Tue 10 Oct (week 4)
     - Due: Wed 25 Oct (week 6) @ 4pm
  2. C programming
     - Out: Tue 7 Nov (week 8)
     - Due: Wed 22 Nov (week 10) @ 4pm

- Must achieve at least 40% in total to pass the course
Late coursework

- Normally, you will not be allowed to submit coursework late

- If you have a **really good reason** to submit late, contact the ITO via their Support Form.
  - The ITO will log the report and pass it on to the UG2 Course/Year Organiser (Dr. Rik Sarkar)
  - Only in exceptional circumstances (e.g., illness that stopped you getting to email), would an extension be granted after a deadline has passed
What is “good reason”?

Something that, in the judgement of the member of staff responsible, would prevent a competent, well-organised, conscientious student from being able to submit on time.

Examples:
- Significant illness
- Serious personal problems

Non-examples:
- Difficult cluster of deadlines
- Last-minute computer problems, dog ate your homework, …
Academic Misconduct (aka Cheating)

- How cheating is defined in this course:
  - Any similarity in the submitted coursework

- How we check for cheating:
  - Pairwise comparisons between all submissions
  - Two tools: MOSS (commercial grade) and an internal tool
    - Rely on sophisticated “fingerprinting” techniques that are rename and reorder proof

- I have a great track record of hunting down cheaters
  - Will go to great lengths to prosecute to the max extent

DON’T DO IT!!!
How to do well in this class

- Get started on assignments early
  - They are intended to be work-intensive

- Take advantage of labs
  - But don’t wait till the last day → demonstrators will be swamped and will not have time for all

- Don’t ignore tutorials
  - Advance prep will allow you to focus on nuances

- Keep up with the reading and the lectures

- Don’t cheat/plagiarise
Well, enough about me! Shall we order?

ZZZZZ
So what is this course about?
Syllabus Overview

- **Hardware:**
  - Data representation and operations
  - Design of (very) simple circuits
  - Processor organisation
  - The memory subsystem
  - Input/Output (I/O)
  - Exceptions and interrupts

- **Software:**
  - Low-level (assembly) programming
  - C programming
  - Operating systems basics
Let’s dig in...
Evolution of computers

- Early computers had their programs set up by plugging cables and setting switches
- John von Neumann first proposed to store the program in the computer’s memory
- All computers since then (~1945) are stored-program machines
Evolution of computers

What has changed is the number of transistors (electronic switches) and their speed. Implementation technology progressed from vacuum tubes to discrete bipolar transistors to (eventually) Integrated Circuits (a.k.a. chips) made with complementary metal-oxide semiconductor (CMOS) technology.
Evolution of computers

- What has changed is the number of transistors (electronic switches) and their speed
- Implementation technology progressed from vacuum tubes to discrete bipolar transistors to (eventually) Integrated Circuits (a.k.a. chips) made with complementary metal-oxide semiconductor (CMOS) technology.
- At the same time, the cost per transistor has been dropping
Moore’s law

date of introduction vs. transistor count

curve shows transistor count doubling every two years
Moore’s law

Transistor counts double roughly every 2 years
- Intel 4004 (1971): 2300 transistors
- Intel Xeon Westmere EX (2011): 2.3 bil. transistors

Million-fold increase in just 40 years!

Biggest CPU today:
AMD Epyc
19.2 Bil transistors
Types of computer systems

- **Servers**
  - Used for either few large tasks (e.g., engineering apps), or many small tasks (e.g., web server, Google)
  - Fast processors, lots of memory
  - Multi-user, multi-program

- **Personal computers**
  - Laptops, desktops
  - Balance cost, processing power
  - Few users, multi-program
Types of computer systems (con’d)

- Mobile devices
  - Smart phones, tablets
  - Highly integrated (multiple processors, GPU, GPS, media accelerators, etc), low-power
  - Single-user, multi-program

- Embedded:
  - Task specific: sensing, control, media playback, etc.
  - Low-cost, low-power
  - Single program
Which computer system category is the largest?

Units shipped per year (millions)

- Servers
- Personal Computers
- Mobile Devices
- Embedded

- Servers: 1 to 10
- Personal Computers: 10 to 100
- Mobile Devices: 100 to 1,000
- Embedded: 1,000 to 10,000
Computer components

- **Data path**
  - Performs actual operations on data

- **Control path**
  - Fetches instructions from program in memory
  - Controls the flow of data through the data path

- **Memory**
  - Stores data and instructions

- **Input/Output**
  - Interfaces with other devices for getting/giving data
A modern processor

- Interface to main memory
- Interface to other chips, peripherals
- Individual processors
- Fast on-chip memory
Modern computer system

- **Compiler**
  - Translates **High Level Language (HLL)** into **machine language** or **byte code**

- **Operating System (OS)**
  - Mediates access to hardware resources (CPU, Memory, I/O)
  - Schedules applications
Words of wisdom & caution

- This class covers a lot of material
  - Keeping up will require effort on your part
  - This ain’t no INF1!
- Attend all lectures, tutorials and labs
- Get started on assignments early
  - They are meant to be work-intensive
- ASK QUESTIONS!

Reward: you will learn a lot!
SLEEP IS FOR THE WEAK