Real Time and Embedded Systems

by
Dr. Lesley Shannon
Email: lshannon@ensc.sfu.ca
Course Website: http://www.ensc.sfu.ca/~lshannon/courses/ensc351

Simon Fraser University

Slide Set: 0
Date: September 6, 2011

Slide Set Overview

- Course Motivation
- Logistical Details
- Course Lab Component
- Lecture Topics

What are Real Time and Embedded Systems?

Real Time Systems Definition

? ? ?

Real Time Systems Definition

- A system that responds in a timely and predictable way to unpredictable external stimuli arrivals.
- Must respond quickly to meet each tasks deadlines
- Often requires simultaneous processing more than one event
  - ***all deadlines (hard, firm, soft) should still be met***
- Predictability/Reliability: react to ***all*** possible events in a predictable way.

Embedded Systems Definition

? ? ?

Embedded Systems Definition

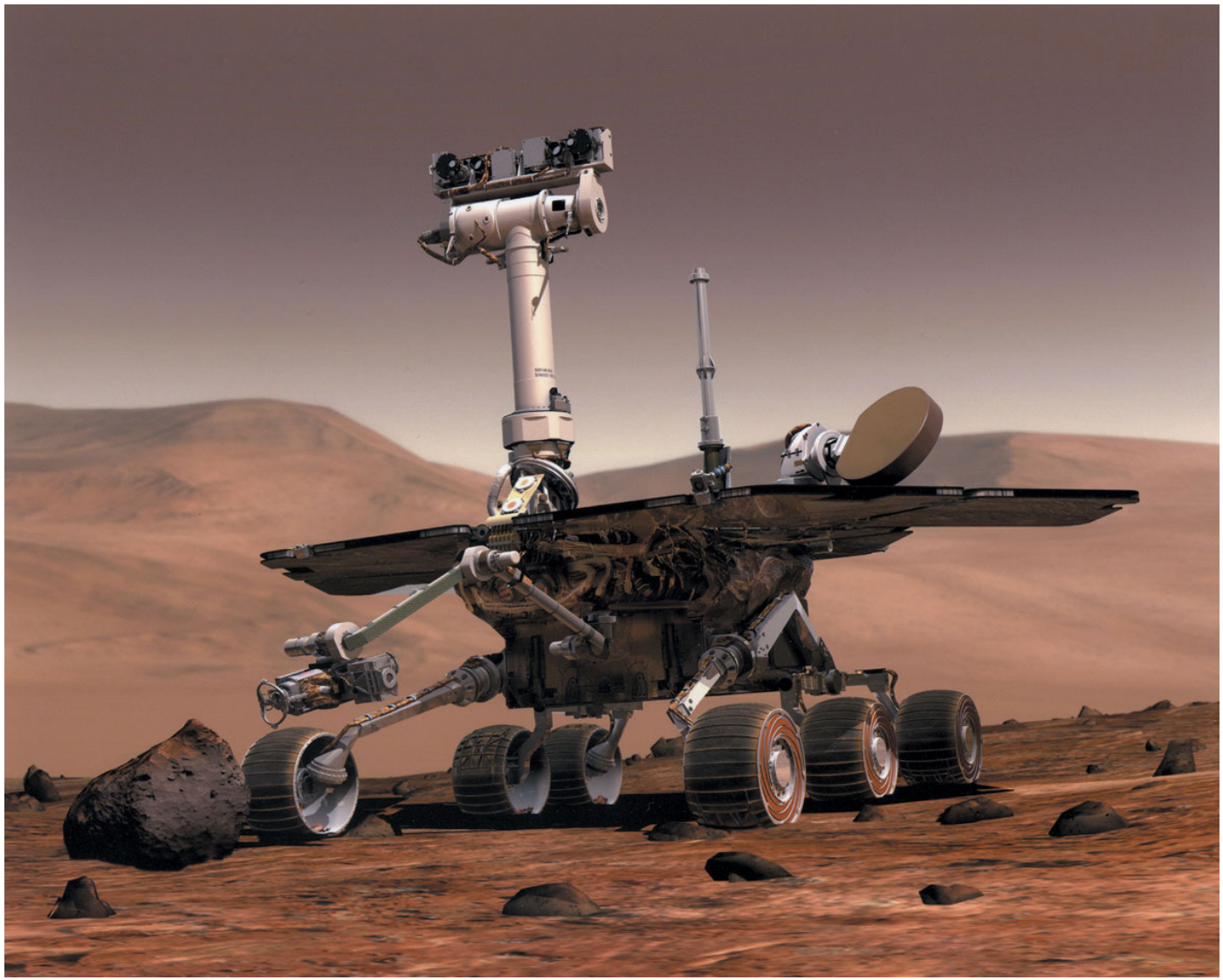
- A computing system that is embedded in a product
- Its primary function is not computing a general computing platform
- Uses a combination of hardware and software to perform the required tasks
  - aka Hardware/Software Codesign
- Typically thought to not be “plugged in”, so there is some alternative power source requirement

Can a system be ***both*** a Real Time System ***and*** Embedded System?

Real Time and Embedded Systems are
EVERYWHERE …

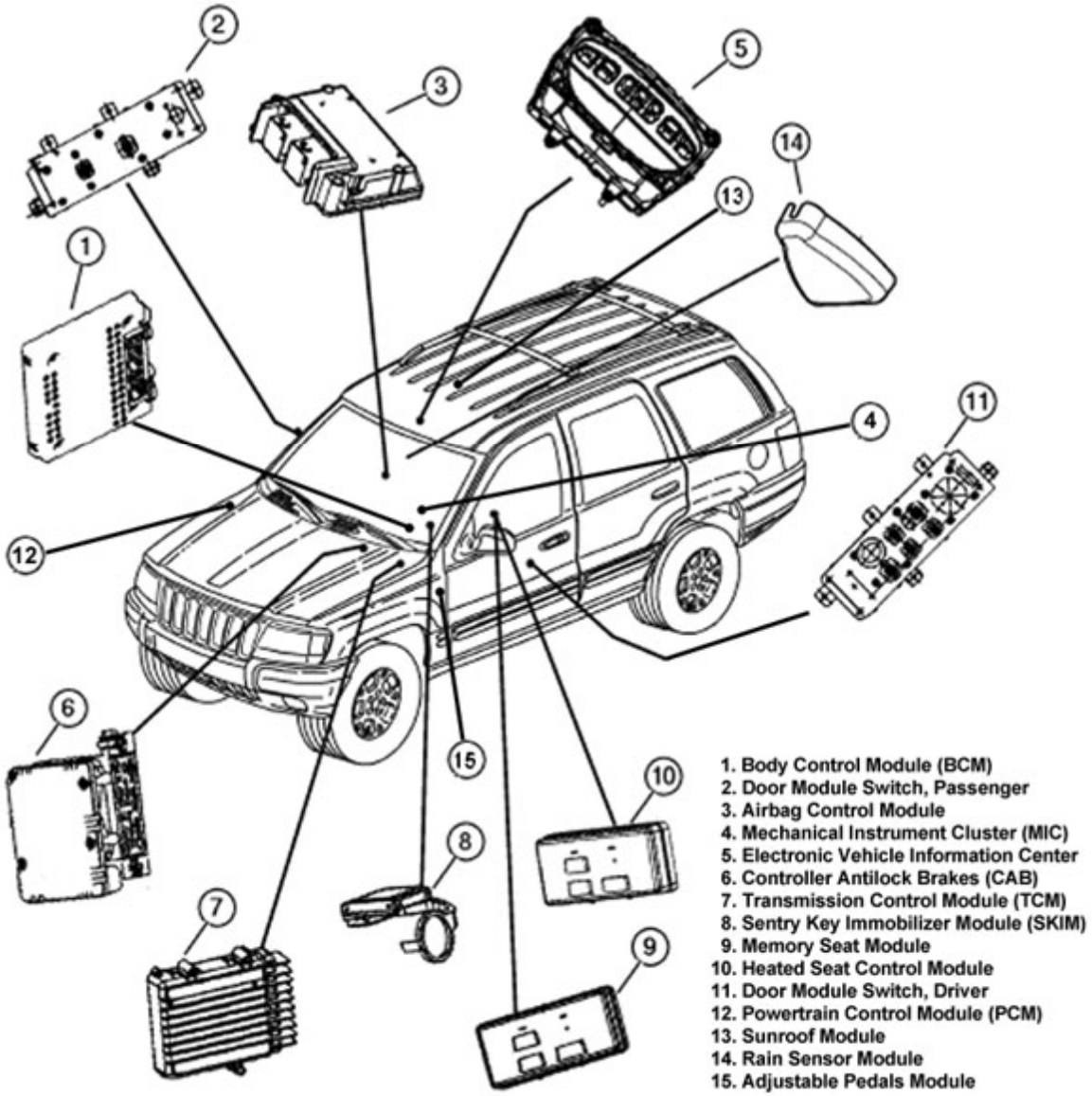
Outer-Space

Mars Rover



On the road

Automotive Electronics



7-Series BMW:
63 Embedded Processors

Mercedes S-Class
65 Embedded Processors

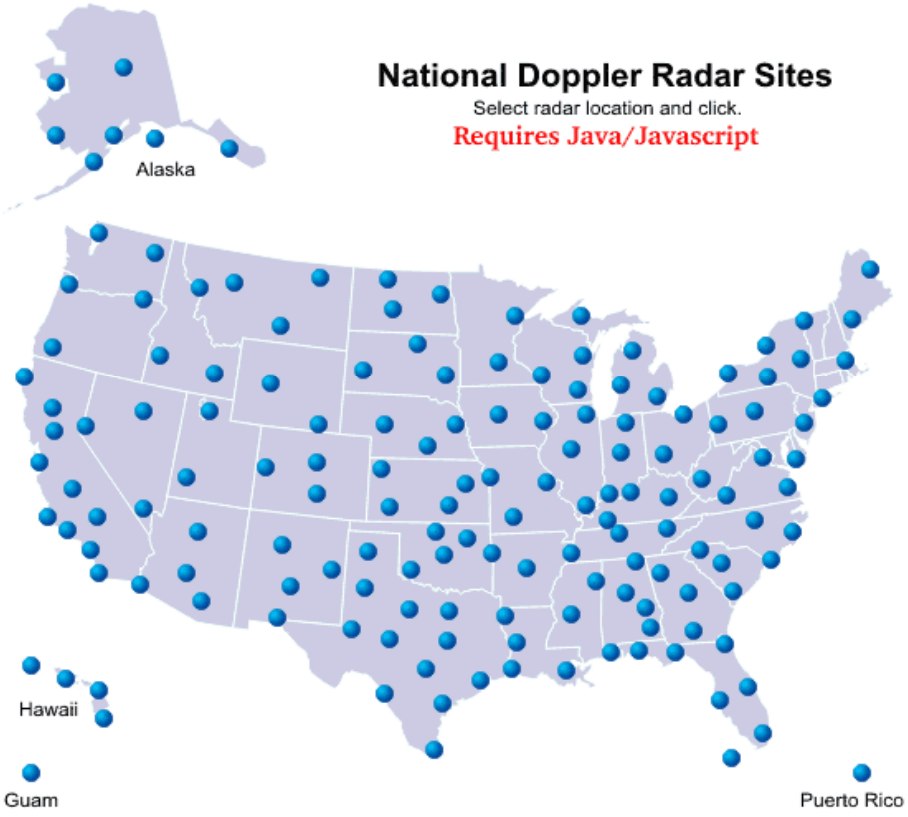
More than 80% of the innovation in autos is from innovations in electronics
- Daimler-Chrysler

Automotive Semiconductor Market: US$13.1 billion / year

Predicting the weather

Meteorological Applications

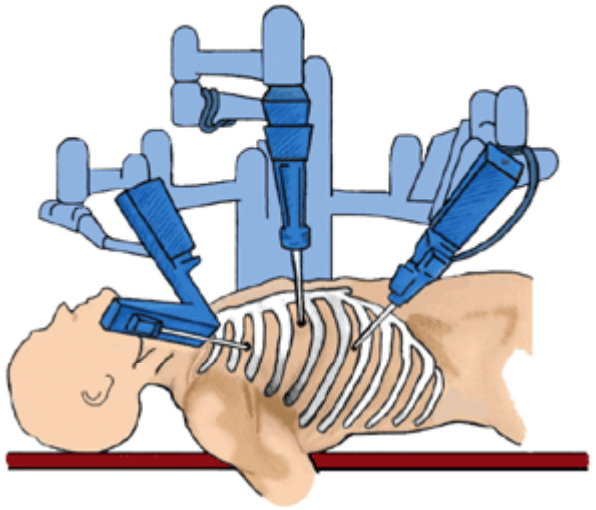
Doppler Weather Radar

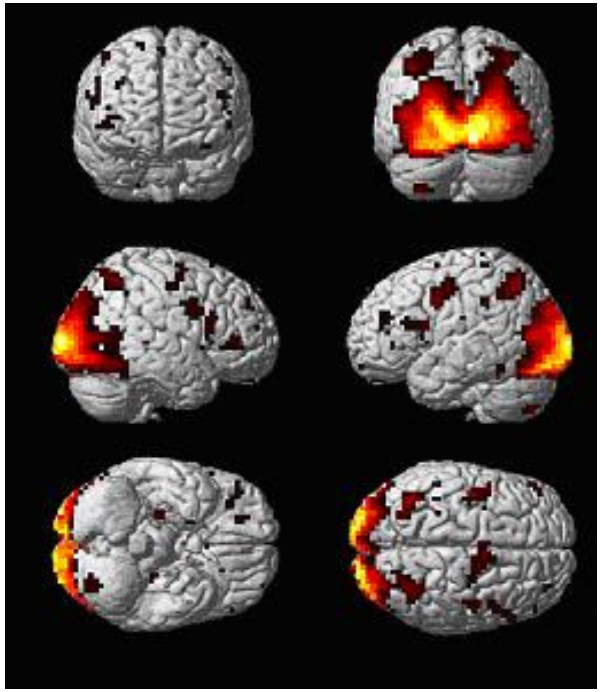


Inside you

Biomedical Applications

Medical Applications





Diagnostic Applications

To build these types of systems, you need specialized software (*and hardware for embedded systems only*) to work in concert (hardware/software codesign)

In this course, we'll only be looking at the ***software*** …

What should you know already?

- Processor Architecture
  - We'll do a crash course review next lecture
- Interrupts
  - We'll do a review later in the course, but you should be familiar with the term
- Assembly Language
  - How does software run on the processor?
  - Why does it matter? (later in course)

What should you know already?

- C/C++
  - We'll be using C
    - A subset of C++ that follows a sequential programming model (not object-oriented)
- Numbers and Arithmetic
  - Base 2 and 16, floating point/fixed point, masking, etc
- I/O
  - Accessing via memory mapping, interrupts, buses, interface circuits, DMA, using peripherals

Why take this course?

- It's "*required*" – a core course for most options

*Note: Two in-class tests.

Why take this course?

- It's "*required*" – a core course for most options

*Note: Two in-class tests.

Why take this course?

- It's ***cool*** !!

- Strong ties between the lectures (theory) and the labs (application)

- Develop practical software system design skills for ***complex*** systems and software team development
  - No "hack" jobs please
  - Try "Extreme Programming" Techniques

Why take this course?

- Embedded Systems are the largest growing computing market.
- "… the computer systems design industry is expected to lead ***all*** service industries in terms of output and employment growth."
  - From: *Looking-Ahead: A 10-Year Outlook for the Canadian Labour Market (2006-2015)*, released by Human Resources and Social Development Canada

What will you learn?

- PetaLinux
  - A full linux kernel for embedded systems (not an RTOS)
- Software Design Issues for Real Time & Embedded Systems
  - Working in large design teams
  - Good software design and testing practices
    - Extreme Programming Techniques

By the end of this course, you should be able to take a customer specification for a complex embedded/real time system:

-create a set of testing criteria and testbenches

-divide the software into separate tasks

-utilize the extreme programming method for software development

-effectively work as a member of a large software design team (use extreme programming)

About Me

- Microelectronics Group: Computer Engineer
- Started in fall ’06
  - I teach ensc ’350, ’351, ’452
- My research group is the Reconfigurable Computing Lab
- My interests include:
  - Reconfigurable Computing, Application-Specific Architectures, Design Methodologies, FPGAs, Embedded Systems, Computing System Design

Reconfigurable Computing Lab

About Me

- One of my major research areas is multicore architecture and effective task scheduling.
  - I use Petalinux in this research project and like to hire research coop students, so if you like this class and do well in it, let me know.

Logistics

My Contact Info

- Office:
  - ASB8819
- Email:
  - lshannon@ensc.sfu.ca
- Homepage:
  - http://www.ensc.sfu.ca/~lshannon/
- Office Hours:
  - Tuesdays & Thursdays:10:30am-12:30pm
- Teaching Assistants (ASB 8803.1)
  - Zia Jalali [Office Hours: Mon, Wed, & Fri 2:30-3:30pm]
  - Eric Matthews [Office Hours: Tues & Thurs 2:30-3:30pm]
  - Kevan Thompson [Office Hours: Mon, Wed, & Fri 10:30-11:30am]

Lectures

- Tuesday 8:30-10:30am AQ3159
- Thursday 8:30-9:30am AQ3003
  - Tutorial: Thursday 9:30-10:30am AQ3003
- 2 in-class pop quizzes (10%)
  - You will receive 3 – 10 days notice
- 1 in-class Midterm (20%)
  - Date: Tuesday, November 1rst
- 1 Final Exam (35%)
  - Date: December 9th, 3:30-6:30pm
- There may also be a Final Exam tutorial (TBD)

Labs

- Lab Test (10%)
  - Scheduled for week of September 26th
    - Sign up for lab pairs on WebCT for oral exam (Sept 26th)
    - Demonstration will take place during your lab demo
  - Tutorial modules:
    - Skeleton code and pdf files with instructions available off the course website
    - Handout in your packet
    - Completed in pairs, but people will be ***tested as individuals***

Labs (ASB 10808)

- Lab Project (25%)
  - Teams of 4, subdivided into groups of 2
  - Multi-phased, with deliverables set for:
    - Oct 12th, Oct 26th, Nov 9th
  - Culminates in Final Demo (Nov 23rd) and Final Report (Nov 28th)
  - More details in the Lab Project handout to come

Lab Equipment

- Each team (quad) will be given login access to one machine
  - But first you need to form a lab group and team

- ***You are responsible for the well being and proper care of the equipment in the lab***
  - ***Destruction/vandalism will result in FAILURE of the class***

The Lab

- To get access to a lab account, you need to email Kevan ***via WebCT*** with your quad info:
  - Email Kevan ***through WebCT*** with your quad info: Full Name, Student Number, and login for ***each*** member
  - Also sign up on the lab sheet on my lab door (ASB 8803.1) This is needed to get your lab equipment
- I know we're covering a lot of details now
  - Don't worry about the project yet (we'll talk mo  out it later)
  - Get started on your tutorials!
- Any questions?

The Lab

- The lab computers have been updated with accounts for the lab teams (quad1, quad2, etc)
  - Although you should work in your lab groups (pairs) to do the tutorials and prepare for the lab test, you will be using these accounts:
    - *REMEMBER: EVERYONE WILL BE TESTED INDIVIDUALLY*
    - ***TELL THE REST OF YOUR QUAD WHEN YOU CHANGE THE PASSWORD FOR YOUR ACCOUNT***

The Lab

- Each lab pair should set up their own subdirectory in the team's workspace to work on the tutorials
  - The oral part of your lab test is scheduled for September 26th
    - You can sign up with your partner for a time slot on WebCT ***after*** you have formed into a quad
  - Your lab demo period will be during your lab demo slot on September 28th
    - Time slots to follow
    - The test format is posted as one of your handouts

Course Webpage

- Course Webpage:
  - http://www.ensc.sfu.ca/~lshannon/courses/ensc351/.
- Course Handouts will be posted here.
- Lecture Slides will also be posted before each class.
- We will be using WebCT for bulletin board postings only
  - Do ***NOT*** send technical email to myself or the TAs
- A Course Mailing List has been created for last minute updates ENSC351
  - You cannot send to it (use the bulletin board)

Course Bulletin Board

- All students ***must*** regularly check the course bulletin board.
- All course related questions ***must*** be posted to the bulletin board.
- You should also post any answers you know
- Postings are set to be anonymous (you can sign your name if you like).
- Announcements and hints regarding the labs and project will be posted here.
- Please observe appropriate bulletin board etiquette (***be respectful***).
- Good citizenship in this course is expected and will be rewarded.

Course Bulletin Board

Please do not use these forums to post any material that is knowingly false and/or defamatory, abusive, vulgar, hateful, obscene, threatening, invasive of a person's privacy, or otherwise in violation of any law. Do not post any copyrighted material unless the copyright is owned by you. I reserve the right to remove any messages posted and to reveal your identity (or whatever information is known about you) in the event of a complaint or legal action arising from any message posted by you.

By posting your message, you agree to indemnify me, my employees, agents and representatives, and to hold them harmless from any and all claims and liabilities (including lawyers' fees) resulting from any material posted to these forums, or from any acts resulting from participants' use of these forums.

Course Bulletin Board

- The different Bulletin Board Categories are:
  - Course Work Related Topics
    - Lectures
    - Administrative details
    - Lab tutorials
    - Projects
  - Uncategorized Topics
    - Default Topic
    - SFU Cafe

Course Reference Material

There is ***no*** textbook this year.

Course Reference Material

- The recommended references are:
  - A good book on C
  - A good book on Linux
    - There are plenty around (on the web even) and I've got some suggestions on the next page.
  - Corbet et al. *Linux Device Drivers* 3rd edition (A free pdf version is available online)
  - Andrew S. Tanenbaum. Modern Operating Systems (3rd edition) Prentice Hall, 2008 (same as CMPT 300)
  - R.J.A Buhr, R.S. Casselman, *Use Case Maps for Object-Oriented Systems*.
  - Extreme Programming:

  http://www.extremeprogramming.org/

Possible Linux Reference Books

- On the kernel:
  - D. Bovet and M. Cesati. “Understanding the Linux Kernel
  - R. Love. “Linux Kernel Development”
  - W. Mauerer. “Professional Linux Kernel Architecture”
- On using Linux:
  - Welsh et al. “Running Linux”
  - W. Ball “Using Linux”
  - E. Nemeth et al. “Linux Administration Handbook”
  - R. Smith “Linux Power Tools”

Course Grade Breakdown

- 10 % Lab Test
- 25 % Lab Project
- 10 % Two Pop Quizzes
- 20 % Midterm
- 35 % Final Exam
- 5 % Class Participation ***Bonus***

Class Participation ***Bonus***

- 5 % Class Participation:
  - This course requires team work, and it will be rewarded:
    - Making and responding to posts on the course bulletin board
    - In-lecture participation
    - Helpng “the team” (fellow classmates) in the lab

Grade Award Breakdown

- > = 90% A+
- 85 – 89% A
- 80 – 84% A-
- 75 – 79% B+
- 70 – 74% B
- 65 – 69% B-
- 60 – 64% C+
- 55 – 59% C
- 50 – 54% D
- < 50% F

Final comment on grades:

- I don't scale
  - Pro: If everyone does well, everyone gets a good mark
  - But there's always the flip side …

Lecture Topics for the Course

Lecture Topics

- Processor architecture review
- An overview of multi-threaded systems
- Processes and threads
- Drawing multithreaded s/w programs (Collaboration Graph Notation)
- Mutexes and semaphores
- Message passing

Lecture Topics

- Timers
- Interrupts
- Priority inheritance
- Resource managers polling and interrupts
- Deadlines and priority
- System reliability (the evil user)

Lecture Topics

- Real time and embedded systems issues
- Using Extreme Programming Techniques for Software Design
- As time permits:
  - Intro to Source Code Control and CVS
  - Compilers and optimizations
  - …

Closing Thoughts

"Programming today is a race between software engineers striving to build bigger and better idiot-proof programs, and the Universe trying to produce bigger and better idiots. So far, the Universe is winning."

- Author **Rick Cook**, *The Wizardry Compiled*