Course Review and Exam Discussion

- review last bits of lecture N1
- review 2006 exam questions Q4(e), Q5
- final examination:
  - details
  - topics
- review of major underlying themes
- outlook for computer systems

- other issues:
  - ANU UltraSPARC T2 confirmed!!!
  - the Random Cache Challenge!

Final Examination

- Wednesday, 18 June, 09:15 – 12:30, MH

  $\longrightarrow$ *please verify date, time and location!*

- 15 minutes reading time
- permitted material:
  - A4 page (one sheet) with notes on both sides (no attachments)
  - dictionary for students with written departmental approval only

  NO calculator permitted! :(
- powers of 2 table AND necessary PeANUt instruction set are given on the exam paper
- same format / structure as last year (write into boxes on exam paper)
- Q1 & Q2 'optional': marks from MSE Q1 & Q2 will be used if better
- some optional parts in Q4 / Q5
- preparation: if you have questions:
  - Discussion Board (potentially 100 people who can help!)
  - my normal Office Hours + COMP2300 contact times (except 11-12 June)

Exam Topics

1. Fundamental Concepts (12 marks)

   number systems, two's complement, floating-point numbers, computer architecture, CPU architecture and functions, basic binary operations, etc.

2. C Language (18 marks)

   understanding and writing C codes, functions, arrays, string handling, etc.

3. Assembly Level Machine Organisation (25 marks)

   PeANUt architecture, assembly language, simple programs, stack, traps, procedure calls, etc.

4. Memory Systems and Modern Machines (20 marks)

   virtual memory, SPARC assembly language, x86/IA32, etc.

5. Operating Systems and Interconnection Networks (15 marks)

   concepts, history, processes, symbol tables and executable structure, file systems, input and output, communications model, network addressing, routing, TCP/IP

   No longer applicable: 2003 Q5(b)(d)(g), 2002 Q5(f), 2001 Q5(c)(d), 2000 Q5(a)(b)(e)(f)

Review of Major Underlying Themes

- abstraction: multiple levels of (increasing) detail
  - layered computer architecture
  - programming languages: MLI, assembly & high-level languages
  - virtual I/O
  - (Unix) files as a 'R/W stream': also can represent directories, devices, memory
  - networks: network access, transport (TCP and IP) and application

  manage complexity, interfaces, support standards
- virtualization: give the *appearance* of a capability or service;

  decouple services from underlying physical resources
  - e.g.

memory	decouple program address from physical memory address
I/O	decouple service (e.g read, write) from device providing it
OS (e.g. Xen)	decouple OS & its services from a physical machine
URLs	decouple web site from machine serving it
networks (e.g. VPN)	decouple logical network structure & services from physical

  - simplicity, flexibility, better resource sharing

Review of Major Underlying Themes (II)

- standardization: allows systems to be reliably constructed from components (of various origins)
  - e.g. C language (ANSI), procedure call conventions (ABIs - application binary interfaces), TCP/IP and application-level protocols, network addressing conventions
  - also in computer architecture: standard components
- caching (memory hierarchy, including virtual memory):
  - blocking of data: *tradeoff* between reducing overhead / unit data and overhead due to fragmentation (loading unneeded data)
    - also occurs in disk access & organization (amortizes cost of positioning head)
- parallelization: pipelining, multiple instruction issue, multicore, clustering (e.g. WWW search servers)
- *tradeoffs* in many kinds of design, e.g. RISC vs CISC
  - decide what situations are most important, and tailor design choices accordingly

Outlook for Computer Systems

- processors: Moore's Law expected to continue for at least another 10 years
  - increasingly aggressive multicore systems (8, 32, 128, . . .)
    - crisis (and opportunity!) in rewrite of applications (parallelize)
  - but the memory wall looms ever higher!
  - more serious still: the power wall! (and the end of 'overclocked' chips)
  - Moore's Law will also enable small, energy efficient chips
    - $\Rightarrow$ increasing prevalence of embedded processors (mobile and ubiquitous computing)
- operating systems: increasing virtualization of all levels of services
- computer networks: increase of scale, complexity and integration
  - ascendancy of grid computing
- green computing:
  - reduce overall power consumption (e.g. 'smart' power-saving modes)
    e.g. UltraSPARC T2 – 'CoolThreads'
  - must recycle the 10^9's of (obsolete) computers – and safely!; also design for recycling e.g. Dell recycling events (Canberra 27/05/07)
- rapidly increasing complexity and also human dependence on these systems!
  - *who will be able to understand them?*