Introduction

COMP3231/9201/3891/9283

(Extended) Operating Systems

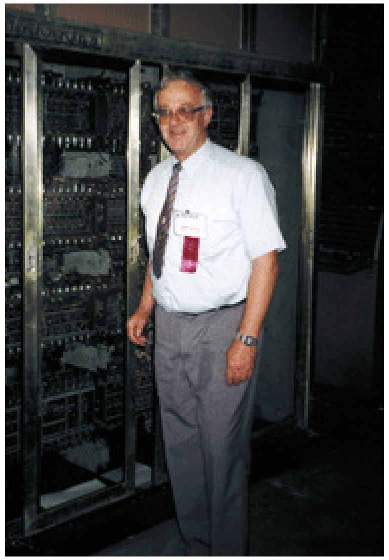
Dr. Kevin Elphinstone

THE UNIVERSITY OF
NEW SOUTH WALES

Operating Systems
@
UNSW

THE UNIVERSITY OF
NEW SOUTH WALES

John Lions (19 January 1937 – 5 December 1998)



- Played a leading role in bringing UNIX to Australia
  - Founding president of Australia UNIX Users Group
- Based his OS course on understanding the UNIX V6 source code
  - Forward thinking at the time.
  - Authored a source code commentary in 1977 to aid understanding

“After 20 years, this is still the best exposition of the workings of a "real" operating system”

— Ken Thompson, co-author of Unix

  - Publication was suppressed by AT&T, and the commentary was widely photocopied "underground".
  - Finally officially published in 1996. Available at http://www.auug.org.au/resources/lions.pdf
  - MIT started an OS course in 2002 based on his commentary
- Lions Garden dedicated in 2002
- 2006 Alumni established *John Lions Chair of Operating Systems*
  - 2009 Gernot Heiser became the inaugural chair.

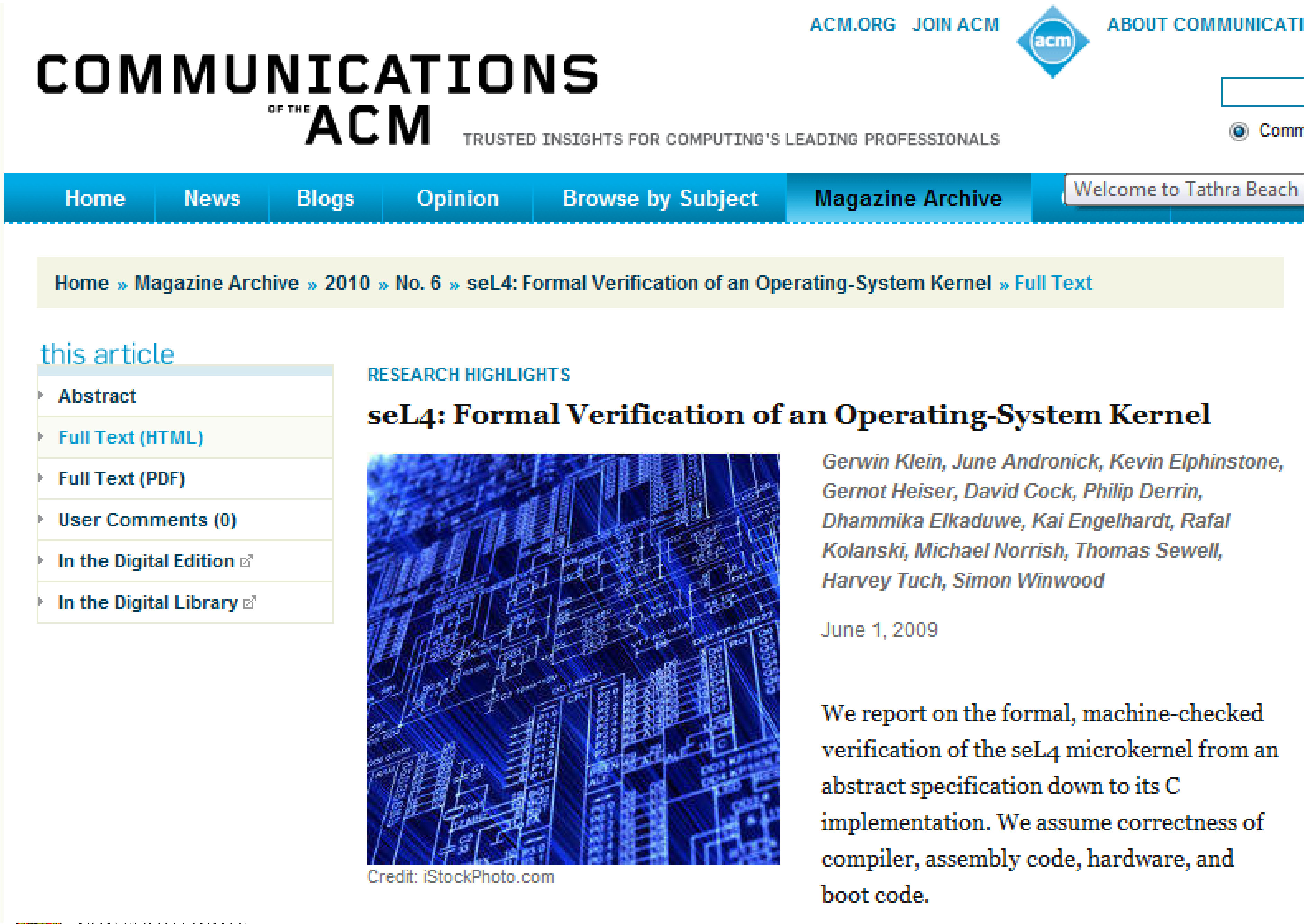
1990s

- 1991 DiSy (Distributed Systems) group started
  - Gernot Heiser (and others) and two PhD students: Jerry Vochteloo and myself.
- 1995 Established collaboration with Jochen Liedtke, original architect of L4 microkernel
  - Developed L4mips microkernel
    - Featured fastest interprocess communication at the time
    - Still fastest on single issue processor
- 1997 COMP9242 Advanced Operating Systems was born.
  - Designed and built U4600:
    - 64-bit MIPS computer
  - Software based on L4mips

U4600

UNSW/NICTA startup OK Labs Timeline

- 1994: Begin of microkernel research at UNSW
- 1997–2003: multiple open-source releases
- 2004: First consulting engagement with Qualcomm
- 2006: Open Kernel Labs founded,
  first L4 phone ships in Japan
- Today: Customer base of blue-chip multinationals
  - Qualcomm, ST-Ericsson, Motorola, …
- Total deployment >1.5 billion devices
- 2012: General Dynamics acquires Open Kernel Labs
- 2014: Wound up



ACM.ORG JOIN ACM ABOUT COMMUNICATI

COMMUNICATIONS OF THE ACM

TRUSTED INSIGHTS FOR COMPUTING'S LEADING PROFESSIONALS

Comm

Home | News | Blogs | Opinion | Browse by Subject | Magazine Archive

Welcome to Tathra Beach

Home » Magazine Archive » 2010 » No. 6 » seL4: Formal Verification of an Operating-System Kernel » Full Text

this article

- Abstract
- Full Text (HTML)
- Full Text (PDF)
- User Comments (0)
- In the Digital Edition
- In the Digital Library

RESEARCH HIGHLIGHTS

seL4: Formal Verification of an Operating-System Kernel

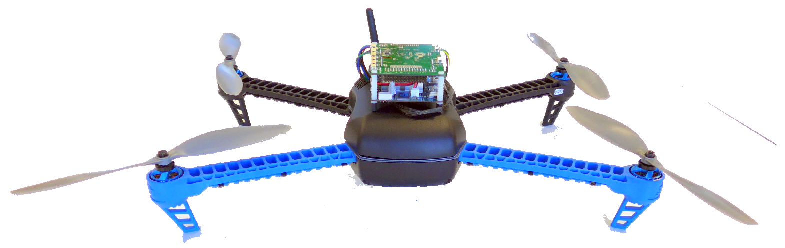
Gerwin Klein, June Andronick, Kevin Elphinstone, Gernot Heiser, David Cock, Philip Derrin, Dhammika Elkaduwe, Kai Engelhardt, Rafal Kolanski, Michael Norrish, Thomas Sewell, Harvey Tuch, Simon Winwood

June 1, 2009

Credit: iStockPhoto.com

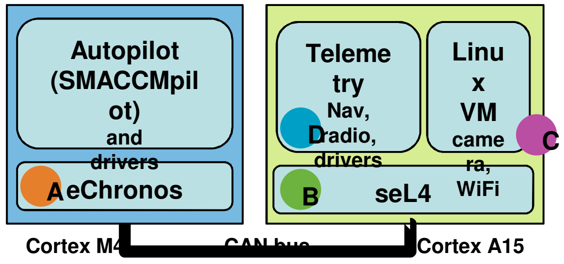
We report on the formal, machine-checked verification of the seL4 microkernel from an abstract specification down to its C implementation. We assume correctness of compiler, assembly code, hardware, and boot code.

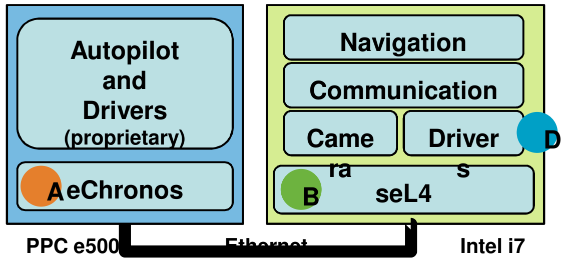
SMACCM: Secure Mathematically Assured Composition of Control Models



Quadcopter Research Vehicle

Unmanned Little Bird





A. eChronos

- Real-time OS
- Verified (in progress)
- Resource limited systems
- No memory protection

B. seL4 & CAmkES

- Verified OS microkernel
- General purpose
- CAmkES component platform

C. Virtualisation

- seL4 hypervisor
- Run legacy applications
- Isolated and protected
- Virtualisation extensions

D. Driver Synthesis

- Synthesise driver code
- Based on device specs
- Correct by construction

DARPA BOEING NICTA galois University of Minnesota Rockwell Collins

Technical Contact: Gernot@nicta.com.au
Business Contact: Jodi.Steel@nicta.com.au

The SMACCM project is part of NICTA's Software Systems Research Group (SSRG), solving fundamental software problems.

Research Excellence in ICT
Wealth Creation for Australia

Welcome to OS @ UNSW

Course Outline

- Prerequisites
  - COMPXXXX Data structures and algorithms
    - Stacks, queues, hash tables, lists, trees, heaps,….
  - COMPXXXX Microprocessor and Interfacing
    - Assembly programming
    - Mapping of high-level procedural language to assembly language
    - Interrupts
  - You are expected to be competent programmers!!!!
    - We will be using the C programming language
      - The dominant language for OS implementation.
      - Need to understand pointers, pointer arithmetic, explicit memory allocation.

Why does this fail?

```
void set(int *x, int *y)
{
      *x = 1; *y = 2;
}
void thingy()
{
      int *a, *b;
      set(a,b);
      printf("%d %d\n",*a,*b);
}
```

Lectures

- Common for all courses (3231/3891/9201/9283)
- Tue, 2-4pm, Webster Theatre B (F Hall B) (K-G15-290)
- Thu, 1-2pm, Chemical Sc M18 (ex Applied Sc (K-F10-M18)
  - The lecture notes will be available on the course web site
    - Available prior to lectures, when possible.
    - Slide numbers for note taking, when not.
  - The lecture notes and textbook are NOT a substitute for attending lectures.
  - Will attempt to have “video” available, baring technical hitches

Extended OS Comp3891/9283

- Thu 2-3pm
  - Old Main Building 150 (K-K15-150)
  - starts in week 2
- A combination of:
  - Examination of topics in more depth
  - Looking at research in area (past/present)
  - OS/161 internals in more depth
- Separate Assessment
  - 75% of final exam common with base course
  - 25% targeted to extended students
  - Advanced assignments part of assessment
- Assumes the tutorials are not challenging enough
  - Effectively replaces the tutorial with extra interactive lecture.

Tutorials

- Start in week 2
- A tutorial participation mark will contribute to your final assessment.
  - Participation means participation, NOT attendance.
  - Comp3891/9283 students excluded
  - Comp9201 optional
- You will only get participation marks in your enrolled tutorial.

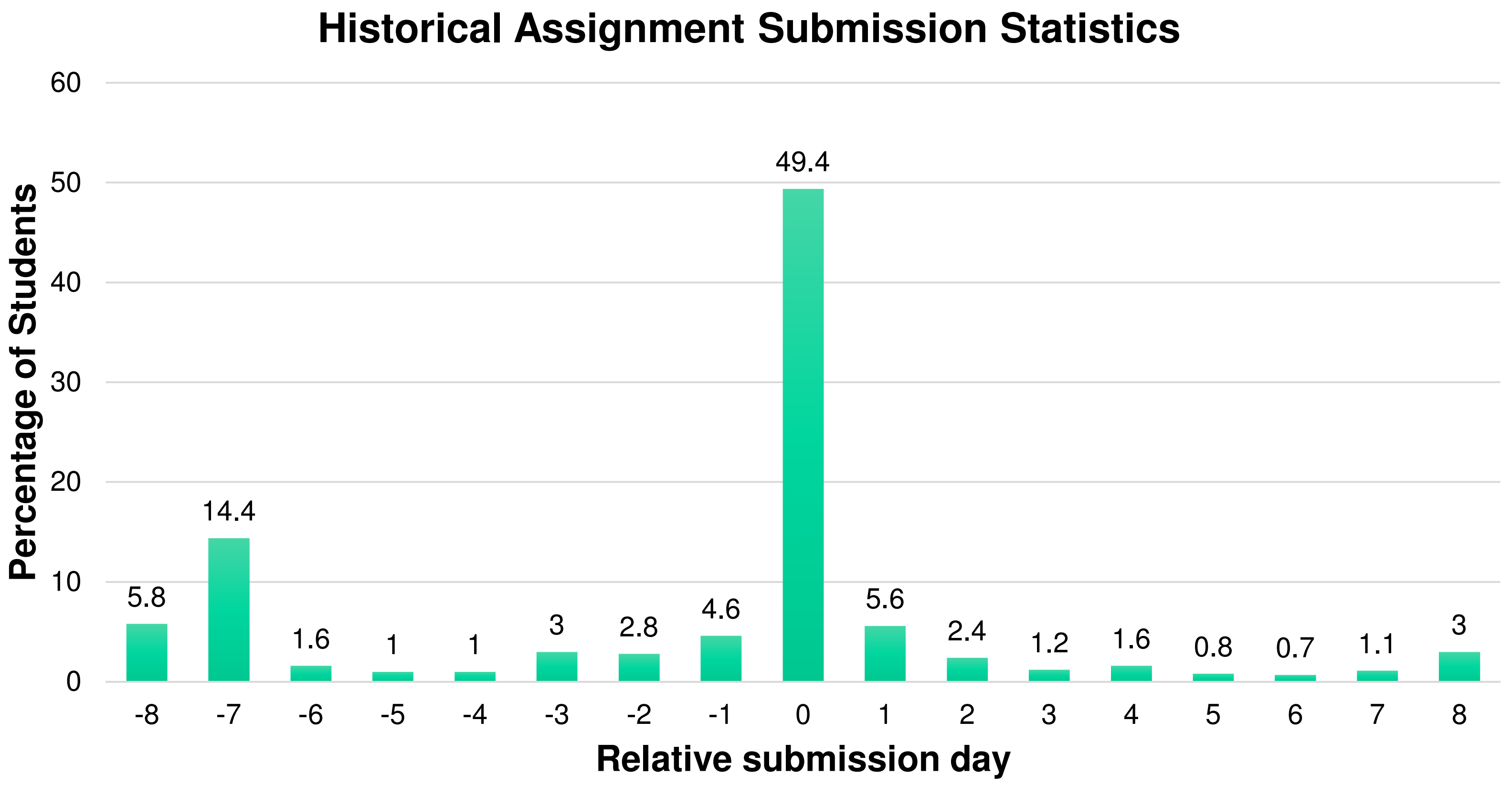
Assignments

- Assignments form a substantial component of your assessment.
- They are challenging!!!!
  - Because operating systems are challenging
- We will be using OS/161,
  - an educational operating system
  - developed by the Systems Group At Harvard
  - It contains roughly 20,000 lines of code and comments

Assignments

- Don’t under estimate the time needed to do the assignments.
  - 80% is understanding
  - 20% programming
- If you start a couple days before they are due, you will be late.
- To encourage you to start early,
  - Bonus 10% of awarded mark of the assignment for finishing a week early
  - See course handout for exact details
    - Read the fine print!!!!

Assignments



16% late

Assignments

- Late penalty
  - 4% of total assignment value per day
    - Assignment is worth 20%
    - You get 18, and are 2 days late
    - Final mark = 18 – (20*0.04*2) = 16 (16.4)
- Assignments are only accepted up to one week late. 8+ days = 0

Assignments

- Assignments are in pairs
  - except warm-up Asst0
  - Info on how to pair up available soon
- Additional, advanced versions of the assignment 2 & 3
  - Available bonus marks are small compared to amount of effort required.
  - Student should do it for the challenge, not the marks.
  - Attempting the advanced component is not a valid excuse for failure to complete the normal component of the assignment
- Advanced assignments part Extended OS student’s (COMP3891/9283) assessment
  - Not optional.

Assignments

- Three assignments
  - due roughly week 6, 9, 13
- Also warm up bonus assignment due in week 4
  - It’s a warm up to have you familiarize yourself with the environment and easy marks.
  - Do not use it as a gauge for judging the difficulty of the following assignments.

Assignments

```
Submission test failed. Continue with
submission (y/n)? y
```

- Lazy/careless submitter penalty: 15%

- Submitted the wrong assignment version penalty: 15%
  - Assuming we can validly date the intended version

Assignments

- To help you with the assignments
  - We dedicate a tutorial per-assignment to discuss issues related to the assignment
  - Prepare for them!!!!!

Plagiarism

- We take cheating seriously!!!
- We systematically check for plagiarised code
  - Penalties are generally sufficient to make it difficult to pass
- We can google as easy as you can
  - Some solutions are wrong
  - Some are greater scope than required at UNSW
    - Makes your assignment stick out as a potential plagiarism case

Plagiarism

- Avoid public github repositories!!
  - From CSE's plagiarism policy
    - Knowingly permitting work to be copied or imitated.
    - Providing an assessment item in full or part to another student to copy, imitate, or produce a derived work.
    - Penalty: Awarded marks are halved.
- Note: bitbucket.org has free academic accounts
  - Unlimited private repositories.

Sample Cheating Statistics

Session	1998/S1	1999/S1	2000/S1	2001/S1	2001/S2	2002/S1	2002/S2	2003/S1	2003/S2	2013/S1
enrolment	178	410	320	300	107	298	156	333	133	108
suspected cheaters	10(6%)	26(6%)	22(7%)	26(9%)	20(19%)	15(5%)	???(?%)	13 (4%)	???(?%)	18 (16%)
full penalties (0FL)	2*	6*	9*	14*	10	9	5	2	1	0
zero for assignment										1
reduced penalties	7	15	7	7	5	4	2	2	9	16
cheaters who failed	4	10	16	16	10	12	5	4	?	1
cheaters suspended	0	0	1	0	0	1	0	0	0	0

*Note: Full penalty 0 FL not applied prior to 2001/S1

THE UNIVERSITY OF NEW SOUTH WALES

Exams

- There is NO mid-session
- The final written exam is 2 hours
- Supplementary exams are oral.
  - Supplementaries are available according to UNSW & school policy, not as a second chance.

3231 Assessment

- Exam Mark Component
  - Max mark of 100
- Based solely on the final exam

- Class Mark Component
  - Max mark of 100
- 10% tutorial participation
  - including optional advanced assignment bonus
- 90% Assignments

3891/9283

- No tutorial participation
- 10% awarded based on advanced assignment attempts
  - Not optional

9201

- Optional tutorial participation, we'll award the better mark of
  - Tutorial participation included as for comp3231
    - Plus any optional advance assignment marks
  - Class marked based solely on the assignments

Undergrad Assessment

- The final assessment is the harmonic mean of the exam and class component.
- If E >= 40,

$$M = \frac{2EC}{E+C}$$

Postgrads (9201/9283)

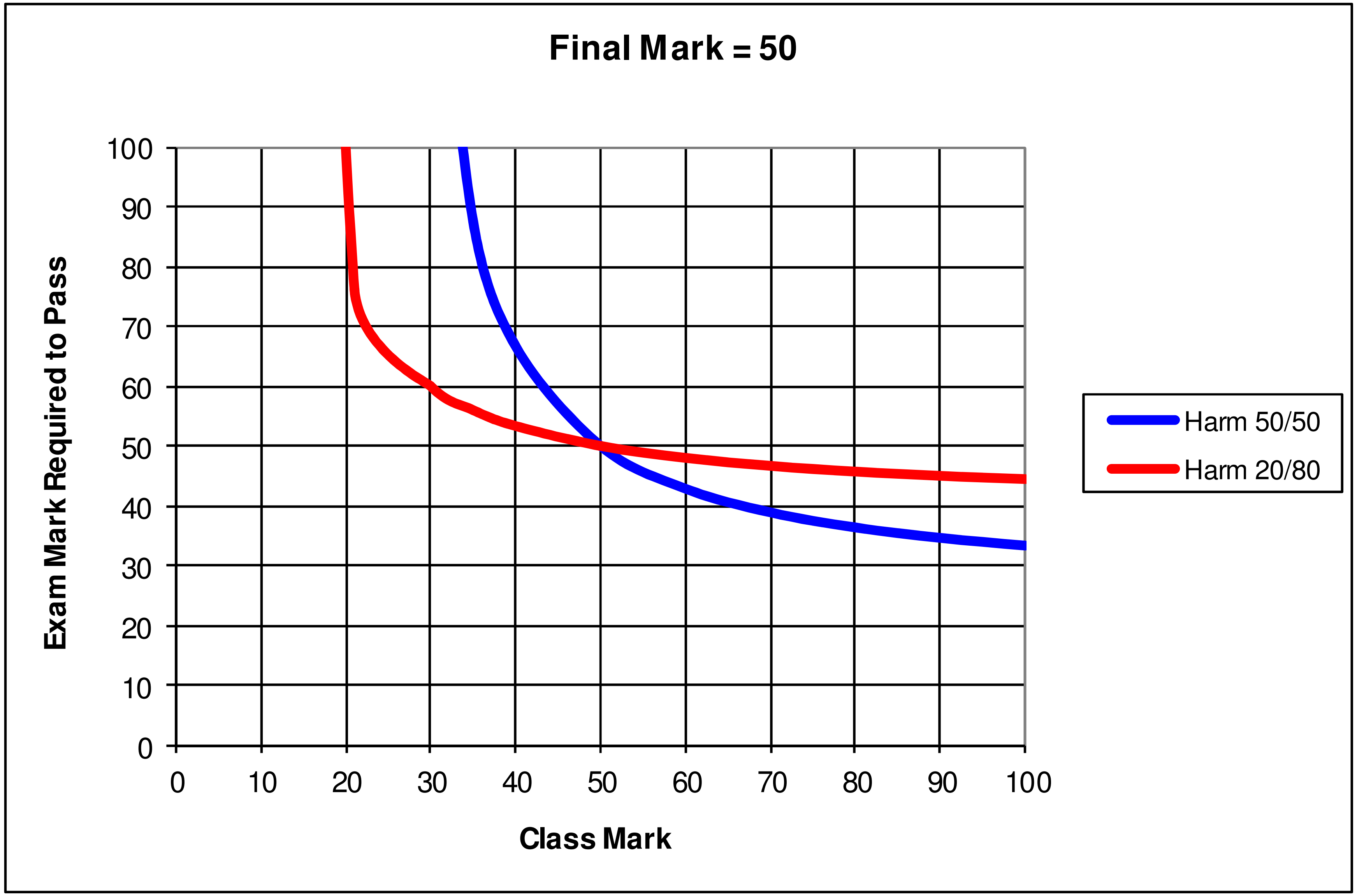
- Maximum of a 50/50 weighted harmonic mean and a 20/80 harmonic mean
  - Can weight final mark heavily on exam if you can’t commit the time to the assignments
  - You are rewarded for seriously attempting the assignments
- if E >= 40,

$$M = \max\left(\frac{2EC}{E+C}; \frac{5EC}{E+4C}\right)$$

Assessment

- If E < 40

$$M = \min\left(44, \frac{2EC}{E+C}\right)$$

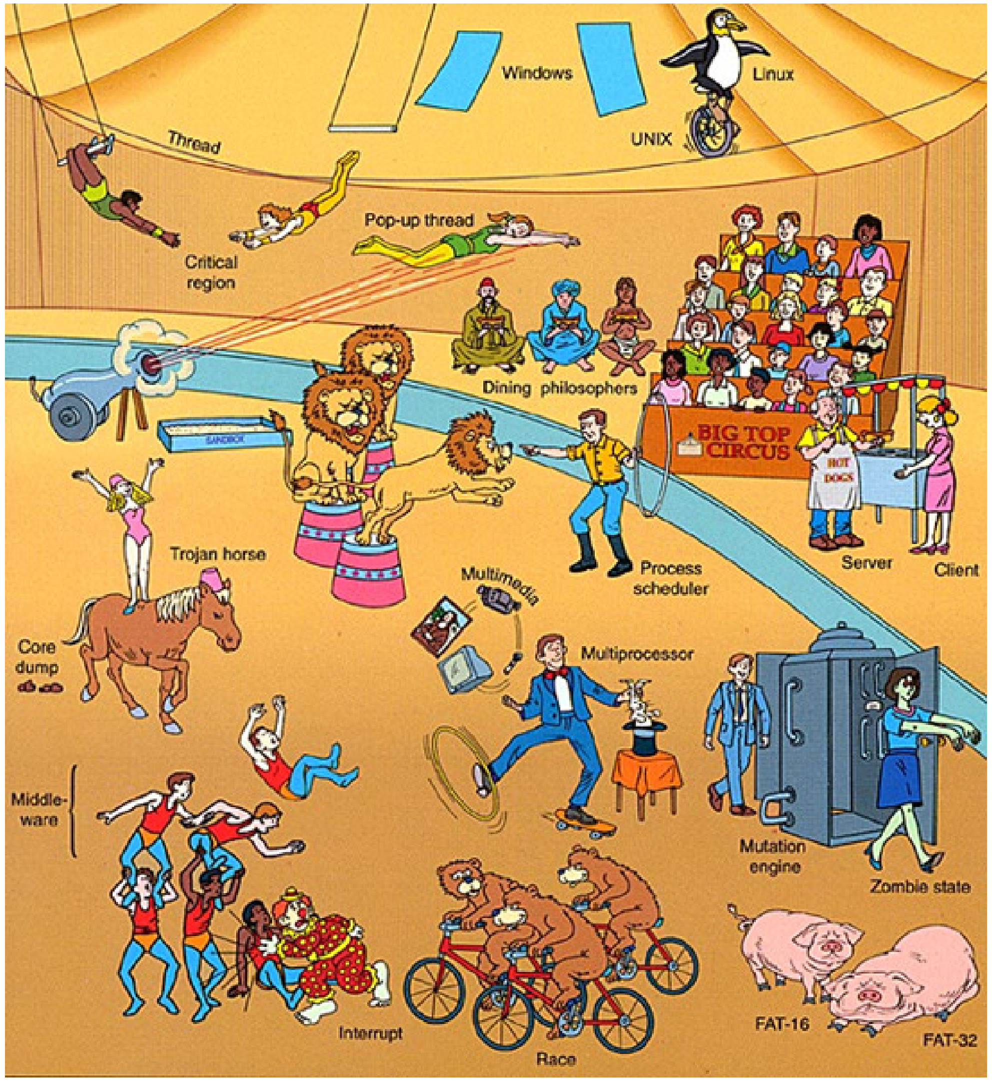

Final Mark = 50
Exam Mark Required to Pass
100
90
80
70
60
50
40
30
20
10
0
0
10
20
30
40
50
60
70
80
90
100
Class Mark
Harm 50/50
Harm 20/80

Assessment

- You need to perform reasonably consistently in both exam and class components.
- Harmonic mean only has significant effect with significant variation.
- Reserve the right to scale, and scale courses individually if required.
  - Warning: We have not scaled in the past.

Textbook

- Andrew Tanenbaum, *Modern Operating Systems,* 3rd/4th Edition, Prentice Hall



References

- A. Silberschatz and P.B. Galvin, *Operating System Concepts*, 5th, 6th, or 7th edition, Addison Wesley
- William Stallings, *Operating Systems: Internals and Design Principles,* 4th or 5th edition, Prentice Hall.
- A. Tannenbaum, A. Woodhull, *Operating Systems--Design and Implementation*, 2nd edition Prentice Hall
- John O'Gorman, *Operating Systems*, MacMillan, 2000
- Uresh Vahalla, *UNIX Internals: The New Frontiers*, Prentice Hall, 1996
- McKusick et al., *The Design and Implementation of the 4.4 BSD Operating System*, Addison Wesley, 1996

Piazza Forums

- Forum for Q/A about assignments and course
  - Ask questions there for the benefit of everybody
  - Look there before asking
  - Apps for phone

- https://piazza.com/
  - Longer link on class web page
    - You will have received an invite from them to you cse email address.
    - Please join and contribute.

Consultations/Questions

- Questions should be directed to the forum.
- Admin and Personal queries can be directed to me kevine@cse.unsw.edu.au
- We reserve the right to ignore email sent directly to us (including tutors) if it should have been directed to the forum.
- Consultation Times
  - See course web site.
  - Must email at least an hour in advance and show up on time.

Course Outline

- "the course aims to educate students in the basic concepts and components of operating systems, the relevant characteristics of hardware, and the tradeoffs between conflicting objectives faced by operating systems in efficiently supporting a wide range of applications."

Course Outline

- Processes and threads
- Concurrency control
- Memory Management
- File Systems
- I/O and Devices
- Scheduling
- Security (maybe)