Welcome to OS @ UNSW

COMP3231/9201/3891/9283
(Extended) Operating Systems
Dr. Kevin Elphinstone

Questions
• Ask any questions you have in the course forum before the first lecture.
• I’ll answer either on the forum or in the first lecture.

System Software Structure

Hardware
Compiled C Code
System Libraries
System Calls
Operating System
System Calls
Compiled C Code
System Libraries
System Calls
Operating System

Major OS Topics

Processes and Threads
Concurrency and Synchronization
File Systems
Scheduling
Multiprocessors

Why Learn Operating Systems?
• Understand the whole software stack
• Develop OS code
• Develop concurrent code
• Application performance
  • Understand operating system behaviour and how best to interface with it.
  • Diagnose system performance issues.
How will we learn about Operating Systems?

- Introduce OS theory and case studies
- Re-enforce theory
- Provide guidance on the assignments

Assignments:
- Opportunity to write real OS code
- OS/161 is a simplified UNIX-clone intended for teaching
- Consist of the following:
  - Warm-up exercise
  - Concurrency and synchronisation
  - OS structure involving system calls and file system
  - Memory management

Intended schedule*

- Lectures
  - Weeks 1-5, 7-9
- Tutorials
  - Weeks 2-5, 7-10
- Assignments Due
  - ASST0 – Week 2
  - ASST1 – Week 4
  - ASST2 – Week 7
  - ASST3 – Week 10

* Subject to change

Overview of Course Outline

Prerequisites
- Data structures and algorithms
  - COMP2521, COMP9024 or COMP1927
  - Stacks, queues, hash tables, lists, trees, heaps...
- Computer systems
  - COMP2521, DSP2100, COMP2121, COMP9052 or ELEC2142
- Computer systems architecture
- Assembly programming
- Mapping of high-level procedural language to assembly language
- Interrupts

Assumed Knowledge
- Computing Theory and Background
  - Basic computer architecture
  - UHD, memory buses, registers, machine instructions, interrupts/interruptions
  - Common CS algorithms and data structures
  - Lists, trees, heaps
  - Basic C programming
  - Reading and writing assembly language
  - Exposure to programming using low-level systems calls (e.g. reading and writing files)
- Practical computing background
  - Capable of UNIX command line usage
  - Familiar with the git revision control system
  - Proficiency in C programming
  - Understanding pointers, pointer arithmetic, function pointers, memory allocation (malloc())
  - The dominant language for OS (and embedded systems) implementation
  - Comfortable navigating around a large-ish existing code base
  - Able to debug an implementation

Why does this fail?

```c
void set(int *x)
{
    *x = 1;
}

void thingy()
{
    int *a;
    set(a);
    printf("%d\n",*a);
}
```
Why does this fail?

```c
void set(int *x)
{
    *x = 1;
}

void thingy()
{
    int a;
    set(&a);
    printf("%d\n", a);
}
```

Lectures

- Common for all courses (3231/3891/9201/9283)
- 2 * 2 hrs each week.
- The lecture slides will be available on the course web site.
  - Available prior to lectures, when possible.
  - Slide numbers for note-taking, when not.
- Lectures will be face-to-face and live streamed simultaneously.
  - Uses Echo360.
  - There is a live chat which will be monitored by a tutor (soon).
  - Recording will be available afterwards as per usual.

Extended OS Comp3891/9283

Starts in week 1

- A combination of:
  - Examination of topics in more depth.
  - Looking at research in areas (past/present).
  - OS/161 internals in more depth.
- Separate Assessment
  - 80%-ish of final exam common with base course.
  - 20%-ish targeted to extended students.
  - Advanced assignment components part of the assessment.
- Assumes the tutorials are not challenging enough.
  - Effectively replaces the tutorial with extra interactive lecture.

Tutorials

- Start in week 2.
- A mix of online and f2f.
- Depends on tutorial you enrolled in.
- Attendance is strongly recommended.
  - but not marked.
- Tutorial questions cover a broad range of examples.
  - Answers available online the week after.
  - Use the tutorial to focus where needed.
  - There is intentionally more questions than can be covered.
  - Review the questions beforehand.

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 System Group At Harvard.
  - With local changes.
  - It contains roughly 20,000 lines of code and comments.
  - Comments are part of the documentation.
Assignments

• Don't underestimate the time needed to do the assignments.
  • 80% is understanding
  • 20% programming
• Avoid
  • 1% understanding
  • 9% programming
  • 90% debugging
• If you start a couple days before they are due, you will be late.
• To encourage you to start early:
  • Bonus 2% of awarded mark per day early, capped at 10%
  • See course outline for exact details
  • Read the fine print!

Assignment Submission Times

Late penalty

• 4% of total assignment value per day
  • Assignment is worth 20%
  • You get 16, and are 2 days late
  • Final mark = 16 - (20*0.04*2) = 16.4
• Assignments are only accepted up to one week late. >5 days = 0

Assignment 0

• Warm-up exercise due in week 2
  • It's a warm-up to have you familiarize yourself with the environment and easy marks.
  • Practice with git revision control
  • Practice submitting a solution
  • Practice using code browser/editor
  • 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!!!!!

Group Work Policy

- Groups of two
- Group members do not have to be in the same tutorial
- Group assignments will be marked as a group
  - Including 'groups' of one.
- Group members are expected to contribute equally to each assignment.
  - No "I'll do the 2nd if you do the 3rd assignment"
  - We accept statements of unequal contributions and do adjust marks of the lesser contributor down.
- Submissions are required to have significant contributions attributable to individual group members.
  - E.g. verifiable using the git revision control system

Plagiarism

- We take cheating seriously!!!
- We systematically check for plagiarised code
- Penalties are generally enough to make it difficult to pass
- We can google as easy as you can
  - Some solutions are wrong
    - You do more than required
    - Makes your assignment stick out as a potential plagiarism case
  - We do vary UNSW requirements

Exams

- There is NO mid-session
- The final written exam is 2 hours
- Supplementary exam are available according to UNSW & school policy, not as a second chance.
  - Medical or other special consideration only

Assessment*

- Exam Mark Component
  - Max mark of 100
  - Based solely on the final exam
- Class Mark Component
  - Max mark of 100
  - 100% Assignments
- * Course outline is authoritative.

Assessment

- The final assessment is a weighted geometric mean of 60% exam (E) and 40% class (C) component.
  \[ M = e^{\frac{60 \ln E + 40 \ln C}{100}} \]
- Additionally, minimum of 40 required in exam (E) and class (C) components to pass.
Assessment

- You need to perform reasonably consistently in both exam and class components.
- Geometric mean only has significant effect with significant variation.
- Reserve the right to moderate marks, and moderate courses individually if required.
- Warning: We have moderated marks only once in the past.

Textbook


References

- Uresh Vahalla, UNIX Internals: The New Frontiers, Prentice Hall, 1996
- McKusick et al., The Design and Implementation of the 4.4 BSD Operating System, Addision Wesley, 1996.

Ed Forums

- Where announcements are posted!!!
- Forum for Q/A about assignments and course
- Ask questions there for the benefit of everybody
- Share your knowledge for the benefit of your peers
- Look there before asking

  - https://edstem.org/
  - Longer link on class web page
  - You will have received an invite from them to your UNSW email address.
  - You need to join to follow the course.

Enforcing standards

- Don’t be offended if we reject your post
- Simply post again following the guidelines

A good example

- Search first! If you have a question and you think you are the first person to have it. Search first!
- Don’t be too general. If you feel your question has been answered before posting it. Search first!
- When posting on Piazza, you have access to the source code you’ve sent.
- More than 20% of Piazza searches return a hits to the code you’ve sent.
- Don’t add bitmap (screenshots). Bitmaps are not searchable so you limit the chances of fellow students finding your post.
- Avoid adding unrelated question to a hot topic because you just happened to be there. It makes it hard for others to find your question.
- Provide some context. Cut-n-paste the error if appropriate, and include the preceding output to provide a chance for others to understand what is going on. Mention the OS/machine/environment you’re using if it’s not clear from the cut-n-paste.
- Mark questions/resolved if they are.
- Don’t leave follow-up unresolved if you have fixed your issue.
- Leave questions unresolved if they are!
- I filter using ‘unresolved’ to find outstanding issues, I won’t find them unless they are marked unresolved.
- You’re very welcome to post if you know the answer to an issue.
- The course staff do not have a monopoly on answers, nor do we monitor the forum 24hrs a day. A quick answer can make somebody’s day (or at least avoid wasting it). A responsive forum can be an awesome resource for the entire course.

A bad example

- Don’t be offended if we reject your post
- Simply post again following the guidelines

A good example

- Search first! If you have a question and you think you are the first person to have it. Search first!
- Don’t be too general. If you feel your question has been answered before posting it. Search first!
- When posting on Piazza, you have access to the source code you’ve sent.
- More than 20% of Piazza searches return a hits to the code you’ve sent.
- Don’t add bitmap (screenshots). Bitmaps are not searchable so you limit the chances of fellow students finding your post.
- Avoid adding unrelated question to a hot topic because you just happened to be there. It makes it hard for others to find your question.
- Provide some context. Cut-n-paste the error if appropriate, and include the preceding output to provide a chance for others to understand what is going on. Mention the OS/machine/environment you’re using if it’s not clear from the cut-n-paste.
- Mark questions/resolved if they are.
- Don’t leave follow-up unresolved if you have fixed your issue.
- Leave questions unresolved if they are!
- I filter using ‘unresolved’ to find outstanding issues, I won’t find them unless they are marked unresolved.
- You’re very welcome to post if you know the answer to an issue.
- The course staff do not have a monopoly on answers, nor do we monitor the forum 24hrs a day. A quick answer can make somebody’s day (or at least avoid wasting it). A responsive forum can be an awesome resource for the entire course.
Consultations/Questions
- Questions should be directed to the forum.
- Admin and Personal queries can be directed to the class account cs3231@cse.unsw.edu.au
- Don’t post private threads in Ed
- 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 (cs3231@cse) at least an hour in advance and show up on time.
    - If we get at least one email, we'll run the consult.

What next?
https://wiki.cse.unsw.edu.au/cs3231/cgi/Checklist

Startup Checklist
- Watch the online intro videos
- Study the questions to give first lecture
- Don’t forget to add the date of first lecture (Thu 06/06/2013)
- Write down your plan for the assignment work (including: review, submit, and in Ed)
- Make sure you have access to the assignment work (including: review, submit, and in Ed)
- Take note of the deadline work (including: review, submit, and in Ed)
- Complete Assignment 3