OS - Retrospection

Tid-bits from course outline

This course is oriented towards exposing students to the essential concepts and issues that underly operating systems and their design.

- **Technical**
  - Make students understand the key concepts and mechanisms of modern operating systems:
    - processes and process management,
    - memory management techniques,
    - on-line storage methods (file systems),
    - concurrency issues,
- **Educational**
  - Make students understand the reasons why operating systems are built the way they are, and what the implications and lessons are for other software systems. Specific learning objectives are:
    - appreciation of design trade-offs and design decisions and their dependence on the target environment;
    - exposure to low-level code;
    - exposure to current trends in operating systems research and development.
- **Professional**
  - The tutorial formats will give students practice in the presentation of solutions to an audience of peers, and will challenge them to critique peer technical presentations. Furthermore, the whole course encourages critical examination and analysis of "standard" solutions.
  - The assignments give students an opportunity to develop skills required to work as a team on a technical project, and the opportunity to work with a substantial body of code created by a third-party.

On-line Course Surveys

- The on-line course survey will be available
  - My one – in addition to CATEI one
- Please make time to do it
  - Please do the CATEI one as well
- Award 2 bonus class marks to everyone who completes ***my*** survey.
  - You will be emailed an invite

Final Exam

- Separate papers for OS (3231/9201) and Extended OS (3891/9283)
- Two Hours
- No examination materials allowed
  - Uni approved calculators okay
- Check the timetable for date

Exam Format

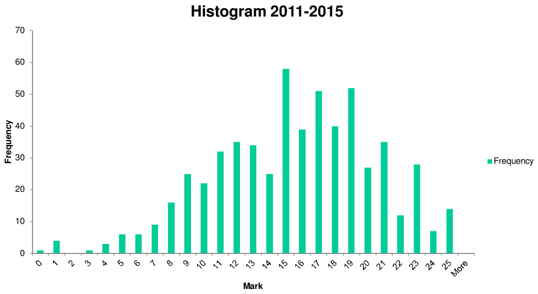
- Read the instructions on the exam
  - The following details are approximate (read the exam instructions on the day)
- 5 questions
  - 3 should be answered in separate books
  - 1 must be ***answered on the exam paper*** itself.
  - 1 must be answered on the multiple choice answer sheet provided
  - 100-ish Marks in total (total will be scaled to 100)
  - 2 marks for following exam instructions

Exam Format

- Q1 is multiple choice (40% marks)

  You will receive one mark for each correct classification, and lose one mark for each incorrect classification. You gain zero marks for each answer left unclassified. The overall mark for this question will not be negative, i.e. the minimum mark is zero.
- Intended to be a mix of easy to hard!
  - Some questions are tricky, and may appear ambiguous if you don't know material.

Q1 mark distribution



Exam Format

- Q2..Q5, roughly:
  - half working out a solution to a problem
  - half written answers to a question

For written answers

- Be clear and concise (get to the point quickly)
  - Long, rambling answers will be penalised

Sample Question

- Name and briefly describe the four conditions required for deadlock. Describe a practical approach to deadlock *prevention,* i.e. a practical approach that prevents one of the conditions described.
- Sample Marking Scheme (out of 10)
  - 2 marks for each condition (1 for the name, 1 for the description), 2 marks for prevention method.

Reasonable answer

Deadlock occurs when threads compete for multiple resources (locks) and the following conditions hold

- Mutual exclusion
  - Threads require access to the resources mutually exclusively
- No pre-emption
  - Once obtained, the resources can't be pre-empted (removed) from a thread until the thread releases them.
- Hold and wait
  - When a thread required multiple resource concurrently, they obtain resource a resource at a time, holding previously acquired resources until all resources are obtained.
- Circular wait
  - The threads are waiting for more resources such that they form a circular dependency chain. i.e. each thread is waiting for a resource the next thread in the chain holds.

A practical deadlock prevention method is to mitigate the circular wait condition by numerically labelling resources, and having a system-wide protocol for requesting resources that acquires any needed resources in numerical order.

Poor answers

- Computers, threads, locks, and pre-emptive scheduling
  - Don't just combine tangentially related stuff you remember
- A prevention method is bankers algorithm.
  - No, bankers algorithm is an avoidance method

Poor answers

- Deadlock is when the system stops. A prevention method is to kill a process when deadlock occurs.
- A deadlock prevention method is to only use a single lock in the system.
- A deadlock prevention method is to only run once process at a time.

Answer the question!!!

- Don't repeat the question, we set the exam, we know what it is!!!!
- Don't just write what you know (or don't know) about the broader topic area
  - You make us have to search for the real answer.
  - You may be correct, but say a lot of unrelated incorrect stuff in the process.
- Don't contradict yourself
  - X is better/faster/more efficient than Y, and later Y is better than X
- Marks are awarded for stating WHY or HOW an answer is correct. We usually explicitly ask for it in the question.
  - Demonstrates understanding

Exam Content

- For structure and style, look at the sample exam from past years.
- For content, the tutorial questions are a reasonable *guide*.
- Wiki has 100-ish sample questions (with student answers).
  - Will also answer questions on the forum
    - sometimes difficult to answer without a whiteboard

The questions attempt to examine understanding rather than particular implementations

- Don't expect
  - "Describe OS/161's exception handling on a timer interrupt"
- But you may get
  - "Describe (in general) a feasible sequence of steps that occur in response to a timer interrupt that results in the current process being pre-empted and a another process running"

Examinable Content

- All Lectures, Tutorials, Assignments.
- More specifically
  - Anything related to learning outcomes