CSC369
Operating Systems

Final Review

University of Toronto, Department of Computer Science

Announcements

- A4 marking will start as soon as grace tokens expire
- During exam period, use Piazza for anything unclear
  - Both questions and answers, works best if you collaborate
- Last tutorial on Wednesday: some exam prep exercises
  - Important to attend!

Overview

- Final mechanics
- Processes & Threads
- Concurrency & Synchronization
- Scheduling
- Memory management
  - Paging
  - Page replacement
- Disk I/O
- File systems
- Deadlock
- Final tips

The Final

- When and where:
- http://www.artsci.utoronto.ca/current/exams/dec17

CSC369H1F	A - E	WED 13 DEC	PM 2:00 - 5:00	GB 303
CSC369H1F	F - L	WED 13 DEC	PM 2:00 - 5:00	SF 2202
CSC369H1F	M - Z	WED 13 DEC	PM 2:00 - 5:00	SF 3202

- Check that you know the room and how to get to it, in advance!
  - www.osm.utoronto.ca/map/
- Make sure to go to the right room!
- Exam starts on the hour, please plan to be there 10-15 minutes in advance!

Final Mechanics

- Final exam is cumulative, all topics are covered
- Slightly more emphasis on topics after midterm but not by much!
  - Topics after midterm: Memory management, file systems and storage management, deadlocks
- Quick answer questions, as well as longer questions that involve more thought, and/or working through solving a problem
  - Based mostly on lecture material, tutorial and lecture exercises
- Closed book, one 8.5”x11” double-sided sheet of notes allowed
  - Just one!
  - **Must include the exam date, time, and room in the upper right corner!**
  - No restrictions on font size
    - A magnifying glass is not an allowed exam aid though! :)
  - **BEWARE: do not just dump text blurbs from your aid sheet on the exam!**
    - **What is the question actually asking? No marks for writing related facts or "buzz words", without answering the actual question**
- No calculators or other aids

Operating Systems (general overview)

- The “big picture”
- The following are just general study guidelines
  - Important concepts and aspects that you should make sure to review
  - Do not ignore everything else, depth is important too

The basics

- What is an operating system?
  - If you cannot answer this .. :)

- Why do we have operating systems? What is their purpose?

Processes & Threads

- What is a process? What is a thread?
- What is the difference between user-level threads and kernel-level threads?

- How are new processes created? Deleted? Zombies?
- What does the address space look like? PCB?
- What states can a process be in?
- How do threads relate to virtual address spaces?

System Calls

- What are the protection domains? Why do we need them?
- How do interrupts work? Why do we need them?
- What happens when a process makes a system call?
- How and when does a context switch happen?

Concurrency

- What is the critical section problem?
- What properties does a solution need to have?
- What is a race condition?
- Synchronization primitives
  - S/W solutions (no H/W support): Peterson's algorithm, Bakery algorithm!
  - Hardware instructions – Test-And-Set, Swap/Exchange!
  - Locks (Spinlocks vs. Sleep locks), Semaphores, CVs, Monitors
- Revisit synchronization problems!
  - Producer-Consumer, Readers-Writers problems
- How would you write a monitor?

Scheduling

- Goals in developing a “good” scheduling algorithm

- Know the properties of different algorithms we discussed
  - FCFS, SJF, RR, MLFQ, etc.
  - Which schedulers are best suited for different workloads?
  - Which schedulers may cause starvation?
  - How can process properties (compute-bound, I/O bound) be used by schedulers?

Memory Management - overview

- Why is memory management important?
  - What are the goals of virtual memory?
  - Why do we have virtual memory if it is so complex?
- What are the mechanisms for implementing mem management?
  - Physical and virtual addressing
  - Partitioning types, paging
  - Page tables, TLB
- What are the overheads related to providing memory management?
- What are the policies related to MM?
  - Page replacement

Virtualizing Memory

- What is the difference between a physical and virtual address?
- What is the difference between fixed and variable partitioning?
  - How do base and limit registers work?
- What is internal fragmentation?
- What is external fragmentation?
- What is a protection fault?

Paging

- How is paging different?
- What are the advantages/disadvantages of paging?
- What are page tables?
- What are page table entries (PTE)?
- What are all of the PTE bits used for?
- Know these terms really well
  - Virtual page number (VPN), page frame number (PFN), offset
- Know how to break down virtual addresses into page numbers, offset
  - And how to translate virtual to physical

Page Tables

- Page tables introduce overhead
  - Space for storing them
  - Time to use them for translation
  - What techniques can be used to reduce their overhead?
- How do linear/multi-level page tables work? (Exercises!)
- Know the terminology : PTE, PDE, PTBR, PDBR, etc.
- You should be able by now to translate manually between hexadecimal, binary and decimal and work with them
  - Hex digits: 4 bits, 0-9, a-f
  - You must be able to do this to do well in the 0x171 exam! ;)
  - Practice! No calculators on the exam

TLBs

- What problem does the TLB solve?
- How do TLBs work?
- Why are TLBs effective?
- How are TLBs managed?
  - What happens on a TLB miss fault?
- What is the difference between a hardware and software managed TLB?

Page Faults

- What is a page fault?
- How is it used to implement demand paged virtual memory?

- What is the complete sequence of steps, from a TLB miss to paging in from disk, for translating a virtual address to a physical address?
  - What is done in hardware, what is done in software?

Page Replacement

- What is the purpose of the page replacement algorithm?
- What application behavior does page replacement try to exploit?
- When is the page replacement algorithm used?
- Refresh these thoroughly
  - Belady's (OPTimal), FIFO, LRU, Clock (second chance)
  - Belady's anomaly, Working Set model, Page Fault Frequency

Advanced Mem Management

- What is thrashing? Possible solutions?
- Multiprogramming correlation with CPU utilization?

- What is shared memory?
- What is copy on write?

- How does the operating system leverage virtual memory to provide these?

File Systems

- Some Topics
  - Files
  - Directories
  - Sharing
  - Protection
  - Layouts
  - Buffer Cache
- What is a file system?
- Why are file systems useful (why do we have them)?

Files and Directories

- What is a file?
  - What operations are supported?
  - What characteristics do they have?
  - What are file access methods?
- What is a directory?
  - What are they used for?
  - How are the implemented?
  - What is a directory entry?
- How are directories used to do path name translation?
- What is a hard link? A symbolic link?
- What's the content of a file/directory/link?

File System Layouts

- What are file system layouts used for?
- What are the general strategies?
  - Contiguous, linked, indexed?
- What are the tradeoffs for those strategies?
  - In what special circumstances might you prefer a method that is not suitable for a general purpose file system? (e.g. contiguous allocation)
- What is an inode?
  - How are inodes different from directories?
  - How are inodes and directories used to do path resolution, find files?
- Refresh A4
  - What's the structure of inodes, directory entries, superblock, etc.?

More FS concepts

- How do we go about building a file system? VSFS..
  - What are the levels of indirection in inodes?
  - Advantages/disadvantages of extent-based approach?
- Performance optimizations
  - What's the file buffer cache, and why do operating systems use one?
  - Why is **buffering writes** useful?
  - What is a **major tradeoff** when it comes to **caching and buffering**?
  - What is **read ahead** and why is it important?

Disk

- Understand the memory hierarchy concept, locality
- Physical disk structure
  - Platters, surfaces, tracks, sectors, cylinders, arms, heads
  - What are some hardware optimizations?
- Disk performance
  - What steps determine disk request performance?
  - What are seek, rotation, transfer?
  - Why try to allocate related data close together?
  - What is FFS, and how is it an improvement over the original Unix file system?

Disk Scheduling

- How can disk scheduling improve performance?
  - What are the components of disk access time?
  - What effect does disk scheduling have on each?
- What are the issues in disk scheduling?
  - Response time, throughput, fairness
- Refresh disk scheduling algorithms
  - FCFS, SSTF, SCAN, C-SCAN, LOOK, C-LOOK

Advanced FS Topics

- Theme: crash consistency, optimizing writes, redundancy

- What are some crash recovery mechanisms?
- What is fsck? What are its limitations?
- How is journaling performed in modern file systems like ext3? Advantages over fsck? Metadata journaling?
- What is LFS? What was the key idea in its design? Advantages and drawbacks?
- Can we handle complete disk crashes? What's the idea behind RAID? Levels of RAID discussed in class?
- SSDs (probably not on the exam though, but highly relevant)

Deadlocks

- What is the definition of a deadlock?
- What are the conditions for deadlock?
- What is Deadlock Prevention/Avoidance/Detection & recovery?
- How does the Banker's Algorithm work? Safe states? (again, revisit all tutorial exercises!)
- What is a resource allocation graph, what is it used for?

- *No questions on concurrent transactions on this topic*

General tips

- Make sure you go over each topic and that you are at the very least able to answer the previous questions
- Course slides, lecture & tutorial exercises, readings
- Review assignments
- Test your knowledge with previous exams
  - Public on the UofT Libraries website – old exams repository
  - Course coverage may have changed since, but not substantially
  - Do not count on just these for prepping though!
- Need help? Use Piazza first! Potentially necessary pre-exam office hours TBA..

Preview:

UNIVERSITY OF TORONTO
Faculty of Arts and Science

DECEMBER 2017 EXAMINATIONS

CSC369H1F

Duration - 3 hours

Aids Allowed:
One double-sided 8.5 x 11 cheat sheet (printed or handwritten)

Student number: _____

UtorID: _____

Last name: _____ First name: _____

Lecture section: L0101 L0201 L2003 (circle only the one you are enrolled in)

---

*Do **NOT** turn this page until you have received the signal to start.*

(Please fill out the identification section above, write your name, UtorID and student#, circle your lecture section, and read the instructions below.)

This final exam consists of 8 questions on 22 pages (including this one and pages that have intentionally been left blank). A mark of at least 40 out of 100 on this exam is required to pass this course. *When you receive the signal to start, please make sure that your copy is complete.*

Answer the questions clearly and legibly. Answers that include both correct and incorrect or irrelevant statements will not receive full marks. Be careful what each question asks! Be specific rather than vague in your answers!

We have provided next to each question, a rough estimate on how much time you should spend on it. These are just guidelines and add up to less than 3 hours. Use your time wisely and answer first the questions that you are confident about, then come back to those you might need more time to think about. Do not panic! Relax and keep focused!

If you use any space for rough work, indicate clearly what you want marked.

Q1:	_____/10
Q2:	_____/16
Q3:	_____/10
Q4:	_____/12
Q5:	_____/10
Q6:	_____/12
Q7:	_____/16
Q8:	_____/14
Total:	_____/100

Good luck!

More General Tips

- Do not panic! Take a deep breath, you've got this!
  - This is your chance to show us what you've learned
  - We WANT to give you the credit that you've earned
- Read carefully!
  - What is the question asking?
  - Don't confuse things, e.g., Lamport's Bakery algo. != Banker's algo.
  - If there's anything unclear, please ask
  - Try not to overthink things, we're not trying to "get you" :)
- Keep track of your time
  - Some questions take more time than others
  - Do not spend too much time on a question if you are stuck – might want to revisit it later

Future of systems

- Exploration
  - Vast area
  - Cutting-edge technology trends
- Inquisitive thinking
  - Don't take things as given, explore the "behind the scenes"
  - Question things
- Hard problems are challenging, but fun!

Concluding remarks

- **Congratulations** on surviving CSC 369 .. so far :)
- It's a tough course, but I hope you found it worthwhile and that it will help you with future job prospects
- Remember course evaluations – please make sure to fill them out!
- Good luck with the final exam!
  - ~~Double-check~~ Triple-check the exam schedule carefully and arrive in advance
- Thank you for a great class, it's been a real pleasure teaching you this term!

The end..