CSE 451: Operating Systems
Winter 2005

Final Review

Steve Gribble

Final Logistics

• Comprehensive: covers entire course
  – lectures, homeworks
  – I won’t ask questions based on projects
• Closed book
• Please don’t cheat
  – no looking at neighbor’s exams
• Tuesday 2:30-4:20pm, in this room
Architectural Support

- Privileged instructions
  - what are they, and who gets to execute them?
  - how does CPU know whether to execute them?
  - why do they need to be privileged?
  - what do they manipulate?
- Events
  - exceptions: what generates them? trap vs. fault?
  - interrupt: what generates them?

OS Structure

- What are the major components of an OS?
- How are they organized?
  - what is the difference between monolithic, layered, microkernel OS's?
    - advantages and disadvantages?
  - which is Linux?
Processes

• What is a process? What does it virtualize?
  – differences between program, process, thread?
  – what is contained in process?
    • what does PCB contain?
  – state queues?
    • which states, what transitions are possible?
    • when do transitions happen?
• Process manipulation
  – what does fork() do? how about exec()?
  – how do shells work?

Threads

• What is a thread?
  – why are they useful?
  – user level vs. kernel level threads?
• How does thread scheduling differ from process scheduling?
  – what operations do threads support?
  – what happens on a thread context switch? what is saved in TCB?
  – preemptive vs. non-preemptive scheduling?
Synchronization

• Why do we need it?
  – data coordination? execution coordination?
  – what are race conditions? when do they occur?
  – when are resources shared? (variables, heap objects, …)
• What is mutual exclusion?
  – what is a critical section?
  – what are the requirements of critical sections?
    • mutex, progress, bounded waiting, performance
  – what are mechanisms for building critical sections?
    • locks, semaphores, (monitors), condition variables

Locks and Semaphores

• What does it mean for acquire/release to be atomic?
• how can locks be implemented?
  – spinlocks? interrupts? OS/thread-scheduler?
  – test-and-set?
  – limitations of locks?
• Semaphores
  – wait vs. signal? difference between semaphore and lock?
  – when do threads block on semaphores? when do they wake?
  – bounded buffers problem
    • producer/consumer
  – readers/writers problem
Process Scheduling

- Long term vs. short term
- When does scheduling happen?
  - job changes state, interrupts, exceptions, job creation
- Scheduling goals?
  - maximize CPU util
  - max job throughput
  - minimize \{turnaround time \mid waiting time \mid response time\}
    - if these are on exam, I will define them for you
  - batch vs. interactive: what are their goals?
- What is starvation? what causes it?
- FCFS/FIFO, SJF, SRJF, priority, RR, MLFQ…

Memory Management

- What good is virtual memory?
- Mechanisms for implementing memory management
  - physical vs. virtual addressing
  - partitioning, paging, segmentation
  - page tables, TLB
- Page replacement policies?
- What are overheads related to memory management?
Virtualizing Memory

- what is difference between a physical and virtual address?
  - fixed vs. variable partitioning?
    - base/limit registers.
    - internal vs. external fragmentation

- paging
  - advantages, disadvantages?
  - what are page tables, PTEs?
    - what are: VPN, PFN, offset? relationship to VA?
    - what’s in a PTE? what are modify/reference/valid/prot bits?

Paging, TLBs

- How to reduce overhead of page table?
  - how do multi-level page tables work?
  - what problem does TLB solve?
  - why do they work?
  - how are they managed?
    - software vs. hardware managed?

- Page faults
  - what is one? how is it used to implement demand paging?
  - what is complete sequence of steps for translating a virtual address to a PA?
    - all the way from TLB access to paging in from disk

- MM tricks
  - shared memory? mmap? COW?
Page Replacement

• what is page replacement algorithm?
  – what application behavior does it exploit?
  – when is page replacement algorithm invoked?
• understand:
  – belady’s (optimal), FIFO, LRU, approximations of LRU, LRU
    clock, working set, page fault frequency
  – what is thrashing? why does it occur and when?

Disk

• Memory hierarchy and locality
• Physical disk structure
  – platters, surfaces, tracks, sectors, cylinders, arms, heads
• Disk interface
  – how does OS make requests to the disk?
• Disk performance
  – access time = seek + rotation + transfer
• Disk scheduling
  – how does it improve performance?
  – FCFS, SSTF, SCAN, C-SCAN?
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 they implemented?
  – what is a directory entry?
• how does path name translation work?

• ACLs vs capabilities
  – matrix
  – advantages and disadvantages of each

FS layout

• what are file system layouts for?
• general strategies?
  – contiguous, linked, indexed?
  – tradeoffs?
• what is an inode?
  – how are they different than directories?
  – how are inodes and directories used to do path resolution, and find files?
FS Buffer cache

• what is a buffer cache?
  – why do OS’s use them?
• what are differences between caching reads and writes?
  – write-through, write-back, write-behind?
  – read-ahead?

advanced topics

• what is FFS, how does it improve over original unix FS?
• how about LFS?