

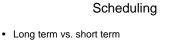
Threads

- What is a thread?
 - why are they useful?
 - what's the address space look like?
 - TCB vs. PCB
 - user-level vs. kernel-level threads?
 - performance implications
 - functionality implications
- 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?

5

- preemptive vs. non-preemptive scheduling?

© 2013 Gribble, Lazowska, Levy, Zahorjan



- When does scheduling happen?
 job changes state, interrupts, exceptions, job creation
- Scheduling goals?
- maximize CPU utilization
 - maximize job throughput
 - minimize {turnaround time | waiting time | response time}
 - batch vs. interactive: what are their goals?
- What is starvation? what causes it?
- FCFS/FIFO, SPT, SRPT, priority, RR, MLFQ...

6

^{© 2013} Gribble, Lazowska, Levy, Zahorjan

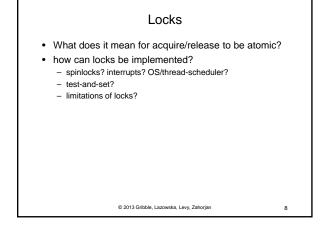
Synchronization

- 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 programming critical sections?
 locks, semaphores, monitors, condition variables

© 2013 Gribble, Lazowska, Levy, Zahorjan

7

9



Semaphores and Monitors

· Semaphores

- basic operations: wait vs. signal?
- difference between semaphore and lock?
- when and how do threads block on semaphores? when do they wake?
- bounded buffers problem
- producer/consumer
- readers/writers problem
- how is all of this implemented
- moving descriptors on and off queues
- Monitors
 - the operations and their implementation

© 2013 Gribble, Lazowska, Levy, Zahorjan

 Understand clearly the queue manipulations involved in implementing semaphores, monitors, condition variables, etc.

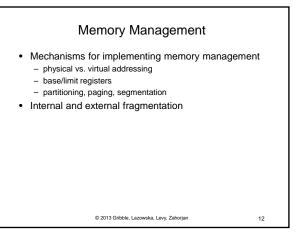
© 2013 Gribble, Lazowska, Levy, Zahorjan

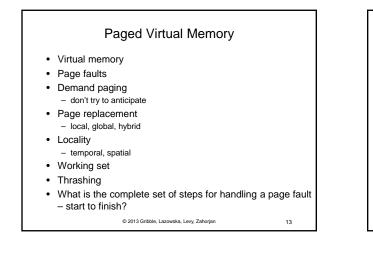
10

Deadlock static prevention, dynamic avoidance, detection/recovery tradeoffs among these graph reducibility approaches Hold and wait Resource ordering Banker's algorithm Detect and eliminate

© 2013 Gribble, Lazowska, Levy, Zahorjan

11





Page replacement algorithms

- Belady's optimal, but unrealizable
- · FIFO replace page loaded furthest in the past
- LRU replace page referenced furthest in the past
 approximate using PTE reference bit
- LRU Clock replace page that is "old enough"
- Working Set keep the working set in memory
- Page Fault Frequency grow/shrink number of frames as a function of fault rate
- VAX/VMS (two-level FIFO due to lack of a referenced bit)

© 2013 Gribble, Lazowska, Levy, Zahorjan

14

16

Multi-level page tables, TLBsHow to reduce overhead of paging?

- 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? Mapped files? copy-on-write?

© 2013 Gribble, Lazowska, Levy, Zahorjan

15

17

Disks

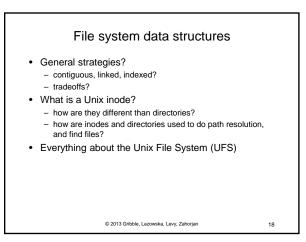
- · Memory hierarchy and locality
- Physical disk structure
- platters, surfaces, tracks, sectors, cylinders, arms, heads Disk interface
- how does OS make requests to the disk?
- Disk performance
- access time = seek + rotation + transfer
- Disk scheduling
 - how does it improve performance?FCFS, SSTF, SCAN, C-SCAN?
- · Implications of solid state drives

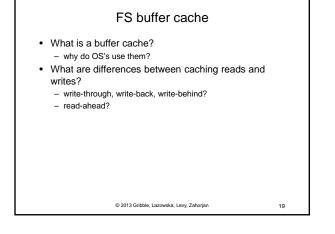
© 2013 Gribble, Lazowska, Levy, Zahorjan

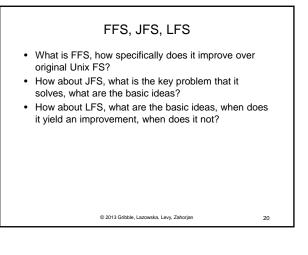
Files and Directories

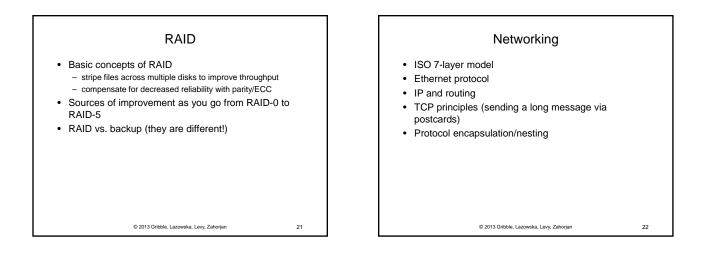
- What is a file
 - what operations are supported?
 - what characteristics do they have?
 - what are file access methods?
- · What is a directory
- what are they used for?
 - how are they implemented?
 - what is a directory entry?
- · How does path name translation work?
- · ACLs vs. capabilities
 - matrix
 - advantages and disadvantages of each

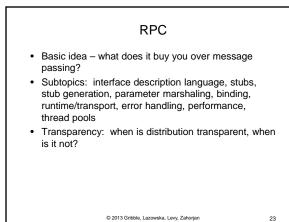
© 2013 Gribble, Lazowska, Levy, Zahorjan

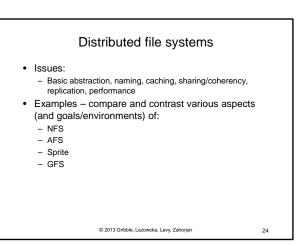








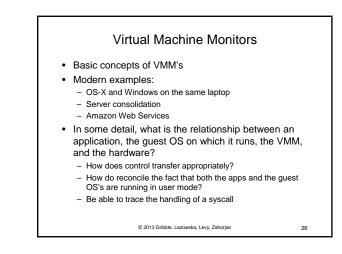


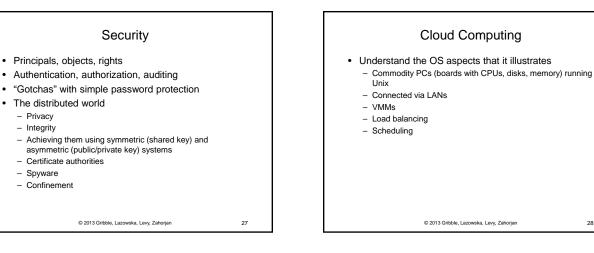


Distributed systems

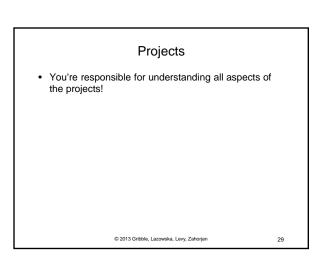
- Loosely-coupled, closely-coupled, tightly-coupled
- Grapevine as an example, in some detail
- Google web search as an example, in some detail
- BOINC
- For Grapevine and Google, focus on reliability, scalability - how do they achieve these properties?

© 2013 Gribble, Lazowska, Levy, Zahorian





25



28