Final Review

The Test

Conditions

- 1:10 PM Wednesday, May 10, in 1127 Mudd
- Open book, open notes, calculator ok
- 170 minutes (but it will probably run short)

Format

- Approximately 15-20 questions
- I'm not asking you to write programs
- Three types of questions, more or less like the midterm's:
  - Explanations of certain concepts, above the pure memorization level
  - Carrying out tasks based on things discussed in class
  - Design questions (i.e., ones intended to make you think)

Material

- If it's in my slides or I said it in class, you're responsible for it
- There will be some material based more on Tanenbaum or other assigned reading; there won't be much from Linux internals
- You're responsible for the assigned readings at about the level of class coverage.
- I'm not going to ask you to memorize the crazy synchronization algorithms, but if one shows up you should be prepared to explain it

CS@CU

Limits

- I can't quiz you on everything I've covered during the semester
- I can't review 30+ hours of class time today
- I'm to some extent limited by the kinds of things it's feasible to ask on an exam

Memory and Virtual Memory

Memory Management

- Why is it needed?
- Relocation
- Position-independent code: used for libraries
- Load-time relocation: good for initial load, but not for realloc
- Hardware assist: base/limit
- Memory protection

Memory Allocation

- Why do we need it?
- Different algorithms: first fit, best fit, worst fit

Virtual Memory

- What it is
- Pages versus page frames
- Page table styles: registers, RAM, segmentation
- TLBs — why they're needed, why they work

More on Virtual Memory

- Process switching
- Fields in a page or segment table entry
- Page faults
- Larger virtual than physical memory

Paging

Page Replacement Algorithms

- Basic problem
- Tools: modified bit, referenced bit, clock interrupts, page faults, advice
- Characteristics of page fault interrupts
- Faking M and R
- Resetting M and R
- How to reclaim pages

Algorithms and Their Properties

- NRU
- FIFO
- Second chance FIFO
- Clock
- LRU
- NFU; aging
- Working set; thrashing

System Issues

- Scheduler interactions
- Anticipatory paging
- Remembering the working set
- Paging disk
- Sharing page tables
- Copy on write
- Locking pages in RAM

Allocation Policies

- Local versus Global policies
- Per-process allocation
- Page fault frequency
- Swapping

Segmentation

- Why use it?
- (Remember that the word is used for two different concepts)
- Properties

Modeling Paging Systems

- Reference strings
- Simulating paging
- Stack algorithms
- Distance strings and page fault rates

File Systems

File Systems

- Types of file systems
- Major design decisions
- Names, hierarchy, extensions, access control, media, versioning, record/block/byte
- Effect of hardware issues

CS @CU

Allocation

- Storage units and types – blocks, tracks, etc.
- Tracking allocated areas

Metadata

- Many types
- Where to store it

Crash Recovery

- Consistency
- Repairing damage
- Log-structured file systems
- Effect of hardware buffering

Unix File System

- Process: working directory and root directory
- Directories, . , and . . , and i-nodes
- Finding a file

I-Nodes

- What's in it?
- Disk blocks and indirect blocks

File Operations

- Opening and closing a file
- Reading, writing, seeking
- Linking and unlinking
- Updating metadata
- Directory operations

Implementing File Systems

- Parts of a file system
- Superblock, i-list, free list

Windows FAT File System

- Metadata and directories
- Freelist implementation
- Long name support

Dump/Restore

- Level 0, 1, etc., dumps
- Dump strategies
- Using the file system versus the physical disk

Permissions

File Permissions

- Classic Unix permissions
- Unix permission-checking algorithm
- "Execute" versus "read"
- Directory permissions
- Deleting files
- Setting permissions; initial permissions

Access Control Lists

- Flexibility
- Order
- Types of permissions

MAC versus DAC

- Who sets permissions?
- Do superusers exist or not?

Other Forms of Access Control

- Temporal
- Access control matrices
- Safety versus security

VFS

- Why a VFS?
- Emulating actions

Device Drivers and I/O

I/O

- I/O models
- Device independence
- The Unix versus the Windows model
- Network devices

Unix Devices

- I/O device names in the file system
- Major and minor device numbers
- Standardized interface
- ioctl

Disk Scheduling

- Seeks, rotational delay
- Demands: fairness, efficiency, promptness
- Elevator algorithm; order of operations

Graphics

- Simple model
- The need for speed
- Programming interface

Network Devices

- Why they're different
- Mux/demux
- Device-dependent issues: framing, address mapping

Physical I/O

- Device quirks
- Interrupt loads and FIFOs
- Programmed I/O versus DMA
- Direct I/O

DMA Issues

- Cache, VM, address space size
- Mapping virtual to physical addresses; where it's done
- Scatter/gather I/O

CS @CU

Error Handling

- Device-specific
- Testing

RAID

- What it is
- Types of RAID

User-Level Issues

User versus Kernel

- What goes where?
- Criteria
- Interfaces

Daemons

- Why?
- Examples: lpr, NFS
- Communications
- Security

Other Commands

- Login
- X
- Shells
- /sbin/init
- Run levels

Security

OS Security

- What is OS security?
- Confidentiality, integrity, availability

Authentication

- Something you know, have, are
- Storing passwords
- Challenge/response
- Tokens
- Biometrics

Attack Techniques

- Trojan horses; spoofing; bugs
- Sandboxes
- Race conditions
- Trusted path
- Viruses and worms

Evaluation

- The Orange Book
- Multi-level security
- Common Criteria
- Features versus assurance
- Protection profiles

Logging

- Why log?
- What to log?

Networking and Distributed OS

Types of Distributed OS

- Multiprocessor, multicomputer, distributed OS
- Memory architectures and speeds
- Latency
- Communications challenges
- Distributed shared memory

Network I/O

- Data copies
- Direct I/O
- Ring buffers

RPC

- What is it?
- Stub procedures
- Marshaling
- The problem of pointers

Network Issues

- Latency
- Reliability, or the lack thereof
- Locking
- Bandwidth

Security

- Identifying and trusting users
- Cryptography
- Capabilities

Distributed File Systems

- Naming
- Performance
- Consistency
- Security

Components

- Devices
- The stack
- Applications

Layers

- (Roughly) their properties
- Interfaces

Sockets

- Network connections aren't files
- Special operations

Applications

- Service
- Servers
- Clients
- Peer-to-peer

Service Applications

- Routing
- Port mapper

Global Grid

- What it is
- Security
- Scheduling
- Sandboxing
- Why it's like an OS