14 - Final Exam Study Guide

CEG 4350/5350 Operating Systems Internals and Design

Max Gilson

Final Exam Details

- 14 questions total - 35% of final class grade
- True/False
  - 4 questions
  - 1.25 points each
  - 5.0 points total (5.0% of final class grade)
- Multiple choice
  - 4 questions
  - 1.25 points each
  - 5.0 points total (5.0% of final class grade)
- Short answer
  - 4 questions
  - 2.5 points each
  - 10.0 points total (10.0% of final class grade)
- Coding
  - 2 questions
  - 7.5 points each
  - 15.0 points total (15.0% of final class grade)

Introduction

- Definitions
  - What is a system
  - What is an OS
- Popular OSs
  - Linux, Windows, MacOS

Design Principles

- Goals
  - User goals
  - System goals
- Separation of mechanism and policy
- High level language vs low level for OS
- OS requirements
  - What does an OS manage?
  - What protections must an OS implement
- Structure
  - Layered, monolithic, microkernel
- Kernel
  - What is a kernel
  - Kernel mode vs user mode
- I/O Access
  - Interrupt, memory mapped, port mapped
- Runtime Stack
  - ESP vs EBP

Hardware and The OS

- Hardware
  - CPU, ISA, RAM, storage, motherboard, etc.
- Boot process
  - PSU, MBR, BIOS, UEFI, booting the kernel
- Shutdown process
- Execution modes
  - Real mode vs protected mode
- Hardware abstraction layer
- Text modes
- Writing to display
  - Character, data, video memory
- Reading from keyboard
  - Status port, data port, scancodes
- Anatomy of storage
  - Cylinders, heads, sections, CHS addressing, LBA addressing

Processes

- Process abstraction
- Multiple processes
  - Context switching
- Parallelism
- Process’s memory
  - Instructions, data, runtime stack
- Inter-process communication
  - Shared files, shared memory, message passing
- Process states
  - New, ready, running, waiting, terminated
- Creating processes in Linux
  - Fork, waitpid, execve, execvp, execlp, exit, kill
- Process control block

Threads

- What is a thread
  - Execution context
  - CPU core vs thread
- Thread abstraction
- Multithreaded vs multitasking
- POSIX
- Kernel threads vs user threads
- Contention scopes
  - PCS, SCS
- Thread models
  - Many-to-one, one-to-one, many-to-many
- Thread scheduling

CPU Scheduling

- Scheduling processes
- Long term scheduling vs short term scheduling
- Scheduling metrics
  - CPU utilization, I/O utilization, throughput, turnaround time, waiting time, response time
- Scheduling algorithms/policies
  - FIFO, RR, SPN/SRTF, MLFQ, lottery
- Calculating waiting time and turnaround time

Main Memory

- Definitions
  - Segment, physical address, virtual address, contiguous memory
- Generating addresses
  - Compile time, load time, run time
- Uniprogramming vs multiprogramming
- Relocation
  - Dynamic vs static
- Allocation
  - Memory holes
  - First fit, best fit, worst fit
- External fragmentation vs internal fragmentation
- Compaction
- Swapping
- Paging
  - OS mapping, paging hardware, pages vs frames
  - Translating virtual address to physical address
- Segmentation
  - Segments, base and limit registers, concepts
  - Translating virtual address to physical address

File Systems

- Directories and files
- File system types (NTFS, FAT, APFS, etc.)
- Goals of a file system
- RAM vs Storage
- Reading/Writing Storage
  - Sectors (unit of atomicity)
  - Read-Modify-Write
- Files as an abstraction
  - Perspectives
  - Metadata
  - File blocks and mappings
- File system performance
- Addressing files
- Storage management
  - Inodes
  - File structure intuitions

FAT12 File System

- Disk organization
  - Boot sector, FAT tables, root directory, data area
- Boot sector format
- FAT table format
  - FAT entry values
- Directory format
  - Directory entries
  - Filename and extensions

File Systems (cont.)

- Allocation methods
  - Extent-based Allocation
  - Linked Allocation
  - Indexed Allocation
  - Multi-Level Indexed Allocation
- Inode Placement
- Hierarchical Unix
  - Naming Conventions
  - Original UNIX format and issues
  - Block size, seek time, cluster solutions
  - Big picture

Input and Output Systems

- Input device vs output device and peripherals
- System architecture
  - System bus, device port, controller, the device
- PCI Bus
- Kernel I/O
  - I/O subsystem, driver, controller
- Port locations
- OS interface
- Polling vs Interrupts
  - How interrupts work steps and interrupt vector table
- DMA
- Device characteristics
- I/O Buffering

Networks and Distributed Systems

- Distributed vs parallel systems
- Advantages of distributed systems
  - Resource sharing, computation speedup, reliability, communication
- Design considerations
  - Communication systems, transparency, security, reliability, performance, etc.
  - Resource sharing, timing, failure recovery, etc.
- Networks
  - LAN vs WAN
  - Packets and routers
  - Protocols
- Network topologies
- Resource sharing
  - Data, computation, and process migration
- Client/server model
- Remote procedure call

Distributed File Systems

- Distributed file system issues
  - Naming, remote access, caching, states
  - Location transparency and independence
- Naming schemes
  - Absolute, mount point, global namespaces
- Accessing files
  - Remote access vs caching
- Caching
  - Where to store cache, how to update cache, cache consistency
- Stateless vs stateful
- Sun’s Network File System (NFS)

Protection

- Objects and access
- Domains and access-rights
  - Domains in UNIX
  - Domains in Multics
- Access Matrix
  - Mechanism and policy
  - Implementations
- Language based protections