Syllabus of Record

Program: CET Siena
Course Code / Title: (SN/CS 3130) Computer Systems and Organization 2
Total Hours: Lecture – 45 hours; Labs – 15 hours
Recommended Credits: 4
Primary Discipline / Suggested Cross Listings: Computer Science / Data Science
Language of Instruction: English
Prerequisites / Requirements: For UVA students: CS 2100 Data Structures and Algorithms 1 and CS 2130 Computer Systems and Organization 1. For students from another institution: prior coursework in Data Structures / Computer Architecture / Programming at same level required.

Description

A second course in computer organization, this course explores a more realistic model of processors and how they and the operating system work together to provide various functionality we depend on as application programmers. Course topics include permission models, system architecture, concurrency, virtual memory, cryptographic primitives, and TCP/IP networking.

Objectives

At the end of this course students will:

- Understand core system architecture, including DMA, interrupts, and caches
- Be exposed to out-of-order and superscalar processors
- Understand concurrency, including synchronization, mutual exclusion, atomic operations, and deadlock
- Understand virtual memory, including process isolation and shared memory
- Understand practical networking, including familiarity with TCP/IP, DNS, DHCP, and TLS
- Be exposed to security, including unix permissions and hardware-based exploits
- Be exposed to ethical and legal issues, including reporting security vulnerabilities
- Use build systems and practical command-line tools

Course Requirements

Active participation is essential in this course. Students are expected to attend each class and field-based course component, as outlined in the CET Attendance Policy. Students are expected to read all assigned materials before the relevant class session and come prepared to participate thoughtfully in class discussions. Graded assignments include:

- Labs: 13 labs; learning exercises that involve discussions, activities, paired programming or other collaborative coding exercises. Credit is for participation.
- Quizzes: 8 quizzes
- Assignments: programming assignments, puzzles, worksheets, and other activities
- Midterm Exam

- Final Exam

.

Grading

The final grade is determined as follows:

- Participation in labs: 20%
- Quizzes: 20% (each quiz worth 2.5% of total grade)
- Assignments: 30%
- Midterm Exam: 15%
- Final Exam: 15%

Readings

Course packet: a selection of digital resources

Outline of Course Content

Topic 1: Software for the Hardware

- Kernel mode vs user mode: motivation
- Implementation: protected instructions and memory, mode bit, mode-switch via exceptions
- Exceptions: interrupts, faults, traps
- Handling: save-handle-restore, exception tables, aborts, Linux system calls
- Exception-like constructs: signals, C library functions,
- Virtual memory: processes, regions of memory, segments revisited, pages, protection
- Page tables: single-level, multi-level, fixed-depth high-arity trees, translation lookaside buffer
- Usage: page swapping, shared memory, process switching, direct memory access

Topic 2: User Accounts

- Per-process permissions
- User accounts
- One account to rule them all
- Daemons
- What is the “operating system”?

Topic 3: Buses and Networks

- Channel characterization: simplex, half-duplex, full-duplex, duplexing
- Network topology: full-connected, star, bus
- “The Bus”: frontside bus, backside bus, memory bus, universal serial bus, SCSI, SATA

Topic 4: Networking Protocols

- Packets with headers: paper, digital, layers
- TCP, IP, UDP
- URLs: the value of mapping, from files to DNS
- DHCP

Topic 5: Security Protocols

- Tools: Hashes, symmetric ciphers, asymmetric ciphers, Diffle-Hellman key exchange, cryptographically-secure random numbers
- Using the tools: signatures, trust, storing passwords, private communication
- Side channels

Topic 6: Caches

- Locality: cache line,
- Types of caches: full-associative, direct-mapped, set-associative
- Common memory caches: L1, TLB, L2- L*n,* write back, write through
- Cache-efficient code

Topic 7: Processes and Threads

- Preemptive and cooperative multitasking
- fork and waitpid functions

Topic 8: Synchronization Primitives and Using POSIX Threads

- Atomic operation
- Semaphore
- Mutex / lock
- Condition variable
- Monitor
- Reader-writer lock
- Barrier
- Transaction
- Managing thread existence: pthread functions, return values, crashing, debugging

Topic 9: Deadlock

- Necessary conditions
- Detection: exact and heuristic detection
- Recovery
- Prevention: static and dynamic approaches

Topic 10: Consistency

- Strict, strong, weak, sequential, and eventual consistency