University of Texas at El Paso
Course Syllabus

COURESE DESCRIPTION

Dept., Number	CS4375	**Course Title**	Theory of Operating Systems.
Approval Date	September 2018	**Course Coordinator**	Eric Freudenthal

CATALOG DESCRIPTION

Process and thread management, concurrency, memory management, processor scheduling, I/O management and disk scheduling, and file management

TEXT BOOK

Remzi H. Arpaci-Dusseau and Andrea C. Arpaci-Dusseau, "Three Easy Pieces," http://pages.cs.wisc.edu/~remzi/OSTEP/

COURSE OUTCOMES

Level Zero: Prior Knowledge

- Virtualization **V0a**: User experience with windowed multitasking systems, **V0b**. Have programmed for "bare metal" in arch1., **V0c**. Authored and debugged programs (nominally using tools and OS services), **V0d**. Used VMware to run linux as guest in computer organization course, **V0e**. Motivation for and implementation of interrupts including atomic save & restore of PC & SR
- Concurrency: **C0a**. How vectorization, pipelining, and locality are exploited to increase throughput of a modern CPU, **C0b** That modern CPUs contain multiple cores and caches
- Addressing: **A0a**. Have typed URLs into browsers, **A0b**. Have organized files into hierarchies
- Encoding, persistence, and communication: **E0a** Stored and retrieved named documents in a hierarchical file system, **E0b**. Written programs that store data in files, **E0c** Used a variety of programs that communicate over the network or store/retrieve data in named files or resource locators, **E0d** Aware of zip, jpg, gif, mp3, but don't know how they work
- Mature programming. **V0a**. Select appropriate data structures, modularization, and user-defined named used within small programs. **V0b**. Written communication: Can compose clear written prose describing or motivating a simple data structure or algorithm.

Level 1: Familiarity

- Virtualization: **V1f**: canonical process state diagram, **V1g** Difference between mechanism and policy, **V1h**. Describe and motivate role of traps, supervisor mode, and memory translation in implementing aspects of LDE (limited direct execution), **V1i**, Describe and motivate core principles of scheduling families. Eg. monoprogramming, prioritized, fairness, fifo, (non) preemptive, quantum, **V1j** Motivations for and gross characteristics of paged & segmented memory allocation strategies, **V1k.** Differentiate between system/vm and library calls, **V1l** Gross security characteristics of virtualization, **V1m**. Components of process/vm context, **V1n** Meaning and relevance of locality, LRU, NRU; relate them to demand loading/eviction as strategy to permit larger virtual memory size, **V1o** Motivation for and gross characterization of trusted computing base
- Concurrency: **C1c**. How asynchrony can simplify design, improve reliability and/or provide speedup, **C1d**. Definitions and conditions related to incorrect behaviors under concurrency including (but not limited to) deadlock and inconsistency, **C1d** Principal of serialization bottlenecks as limiting factor in achieving speedup from concurrency. (generalized Amdahl's law)
- Addressing: **A1c**. Principles underlying and gross characteristics of hierarchical, uniform, and non-uniform address spaces, **A1d**. P2P architectures (flood, structured), **A1e**. Motivations for, principles, underlying structure, and scalability implications of segmented & paged memory; domain names, ports, IP and MAC.
- Encoding, persistence, and communication: **E1f**. Motivations and gross characteristics of stream and datagram transport models, **E1g**. Forwarding, E1h. Define and compute simple transmission and propagation latencies, **E1i** How data is serialized (byte order, representation, marshalling), **E1j**.

Lossy v. lossless compression, **E1k** Sharding with regards to distributed data stores, **E1l** Inspection tools (fs editor & packet capture), **E1m** Core ideas of ARQ, flow, and rate control, **E1n**. Gross characteristics of (a)symmetric crypto including key exchange; protocol security properties, **E1o**. Motivations and scalability of inter-address translation strategies (e.g. ARP, DNS. IP masquerading, subnet-driven IP routing), **E1p**. Layering and end-to-end principles, as applied to eth and IP, **E1q**. Data structures and their semantic/performance implications for inode, fat, iso filesystems, **E1r**. Timing and reliability characteristics of common random access mass storage devices; how redundancy can be used to increase reliability and performance, **E1s** Motivations for key aspects device driver design including (1) generic interfaces and (2) kernel/interrupt "halves"

Level 2: Application

- Virtualization: **V2p**. Choose appropriate container family (e.g. OS v. hw virtualization sandbox), **V2q**. Determine and motivate mapping between common process activities and canonical state, **V2r**. Can characterize policies for common LDE contexts, **V2s**. Characterize semantics and identify algorithms/data structures suitable for various memory allocation strategies, **V2t**. Can create and communicate among virtualized execution containers, **V2u** Can construct programs using system and library interfaces.
- Concurrency: **C2f**. Use standard coordination primitives such as mutex to ensure a correctness property of a threaded program, **C2g**. Producer-consumer, **C2h**. Construct a program that correctly responds to messages from multiple communication sources, **C2i** Deadlock, conditions under which it can occur, and standard approaches to avoid it.
- Addressing: **A2f**. Can motivate the common uses and challenges of multiple of addressing schemes in the context of storage, memory allocation, and network addressing, **A2g**. Can perform simple arithmetic computations related to major families (e.g. determine page number or whether an address is within a power-of-2 segment)

Level 3: Analysis and synthesis

- Virtualization: **V3v**. Distinguish between policies and mechanisms in LDE context, **V3w**. Select (or diagnose) gross characteristics of relationship between scheduling policy and behavior, **V3x**. Can analyze/select gross aspects of relationship between addressing family and application context, **V3y**. Analyze whether a function would be better implemented as a system or library call, **V3z** Security implications of various sandboxing strategies.
- Concurrency: **C3j**. Distinguish when blocking vs nonblocking calls are appropriate, **C3k**. Utilizing appropriate data structures for achieving synchronicity in a given problem
- Addressing: **A3h**. Can select/analyze suitability of some addressing scheme for a simple problem

ABET STUDENT OUTCOMES MAPPING

Course outcomes	Student outcome
A2f-g, A3h	1
V3v-z, C3j-k, A3h	2 (ABET 1)
V3v-z, C3j-k, A3h	3 (ABET 2)
None	4 (ABET 5)
None	5 (ABET 4)
M0b	6 (ABET 3)
None	7
None	8
C2f-i	9
None	10 (ABET 6)

PREREQUISITES BY TOPIC

CS 3432 with a grade of C or better