CS 4375: Theory of Operating Systems

Spring 2021

Instructor:  Nigel Ward, nigel@utep.edu
Office Hours: TTh, 2-3, or by appointment,
in Blackboard for now; perhaps later in Computer Science 3.0408, also 747-6827

TA:  Peter Hanson, pghanson@miners.utep.edu
Office Hours: MW, 11–11, or by appointment,
in Blackboard for now, perhaps later in CCSB G.0512

Class Time:  TTh 10:30 – 11:50 in Blackboard

Introduction to Computer Networks, by Peter Dordal.
also recommended: Modern Operating Systems by Tannenbaum and Operating Systems Concepts by Silberschatz.

Course Description:  Process and thread management, concurrency, memory management, processor scheduling, I/O management and disk scheduling, and file management.

Goal:  Introduce concepts that will be foundational for further study, whether academic or professional, of 1) Computer Security and Forensics, 2) Systems Administration and Network Administration, and 3) Systems Programming and Network Programming, including developing for embedded systems, cloud systems, and high-performance systems.

Course Policies

The prerequisite for this class is CS 3432 with a C or better.

Assigned readings are to be done before class.

Assignments are to be submitted in Blackboard, unless otherwise specified. The maximum points attainable for late assignments will be reduced by 10% per day or partial day of lateness, for up to five days, or more if the lateness impedes participation in class or the assignment is received after the solution has been discussed. Email submissions of assignments are not accepted unless otherwise specified.

Assignments are to be done individually unless specifically designated as group assignments. While you may discuss assignments with others, your solutions should be designed, written/assembled, and tested by you alone. If you need help, consult the TA or the instructor.

The use of found code and shared code is acceptable, unless otherwise specified, provided that you acknowledge your sources, state specifically what you used, and understand every line of code.

Programming assignments will be graded primarily on functionally, design quality, thoroughness of testing, and readability. Some of these factors inevitably involve subjective judgments; if you have questions about these or any other aspect of the grading, please see the TA or the instructor.

Tests will be closed-book, except that one single-sided page of hand-written notes may be used for the first test, two for the second test, and three for the final. Use of Lockdown Browser is required.

Grades will be based on four components, weighted approximately as follows: 45% assignments, 22%
final examination, 28% tests, and 5% other factors, including quizzes, in-class exercises, and participation.

Assignments and tests will be challenging. Grading will be on a points-earned basis (points above zero), rather than a points-off basis (points below expectation). Letter grades will be assigned accordingly: the A/B break will probably be around 80% and the B/C break around 70%. Final grades may be adjusted upwards in cases where performance on both the assignments and the tests is solid. Per-assignment grades will be found in Blackboard; overall grades will be communicated separately.

Students are free to attend class or not, bearing in mind that absence may annoy other students, interfere with learning, and result in a lower grade.

**General Policies**

No make-up exams or assignments will be given except under the conditions set forth in the Catalog. Students are expected to be punctual, and, as always, to conduct themselves professionally and courteously.

If you have or suspect a disability and need accommodation, contact the Center for Accommodations and Support Services at 747-5148 or at cass@utep.edu or visit Room 106 Union East.

**Topics, Readings and Major Assignments, tentative**

<table>
<thead>
<tr>
<th>Introduction (3 days)</th>
<th>ADAD Ch. 1-4</th>
</tr>
</thead>
<tbody>
<tr>
<td><em>Python warm-up assignment</em></td>
<td></td>
</tr>
<tr>
<td>Processes (3 days)</td>
<td>ADAD Ch 5-7, 11</td>
</tr>
<tr>
<td><em>Shell assignment</em></td>
<td></td>
</tr>
<tr>
<td>Memory (3 days)</td>
<td>ADAD Ch 12-18</td>
</tr>
<tr>
<td><em>sundry exercises</em></td>
<td></td>
</tr>
<tr>
<td>Test 1</td>
<td>various readings</td>
</tr>
<tr>
<td>Sockets (3 days)</td>
<td>various readings</td>
</tr>
<tr>
<td><em>Sockets assignments</em></td>
<td></td>
</tr>
<tr>
<td>Threads and Concurrency (5 days)</td>
<td>ADAD 25 - 26; 28 - 32, portions</td>
</tr>
<tr>
<td><em>various assignments</em></td>
<td></td>
</tr>
<tr>
<td>Test 2</td>
<td>ADAD 48; Dordal Ch. 1, 9, 17-18, 28</td>
</tr>
<tr>
<td>Networking (5 days)</td>
<td><em>sundry exercises</em></td>
</tr>
<tr>
<td>Files and Devices (4 days)</td>
<td>ADAD Ch. 35-37, 39-40; 38-46 portions</td>
</tr>
<tr>
<td><em>sundry exercises</em></td>
<td></td>
</tr>
<tr>
<td>Final Exam</td>
<td></td>
</tr>
</tbody>
</table>

**Course Website:** [http://nigelward.com/os/](http://nigelward.com/os/)

**Important Dates** (tentative)

- January 19: Class begins
- February 18: Test 1
- March 15-19: Spring Break
- April 1: Test 2
- May 7: Dead Day
- May 13: Final Exam, 10:00-12:45
Outcomes (Study-Guide Version)

Level 1: Knowledge and Comprehension

Level 1 outcomes are those in which the student has been exposed to the terms and concepts at a basic level and can supply basic definitions. The material has been presented only at a superficial level.

Upon successful completion of this course, students will be able to:

V1i. Choose a scheduling approach suitable for given simple problem.
V1j. Explain segmentation and its security implications.
V1k. Explain some ways in which virtualization creates vulnerabilities.
V1l. Explain the components of process and virtual machine context.
V1m. Explain the need for paging, and basic strategies and their possible performance implications.
V1n. Describe the motivation for and gross characteristics of a trusted computing base.
C1c. Given an application, identify the factors relevant to choosing a synchronous or asynchronous solution.
E1f. Choose when to use datagram versus virtual-circuit communication.
E1h. Define and compute simple transmission and propagation latencies.
E1i. Explain how data is serialized (byte order, representation, buffering).
E1j. Explain the difference between lossy and lossless compression.
E1l. Interpret the output of a packet capture tool.
E1n. Explain the role of cryptographic hashes and symmetric and asymmetric keys in security.
E1o. Explain the basic concepts of DNS and IP.
E1p. Explain the functionality handled at different network layers.
E1q. Explain some concepts in storing files on disk.
E1r. Explain the memory hierarchy and the basic concepts of distributed storage.
E1s. Explain generic device APIs, including the bidirectional handling of interrupts and requests.

Level 2: Application and Analysis

Level 2 outcomes are those in which the student can apply the material in familiar situations, e.g., can work a problem of familiar structure with minor changes in the details.

Upon successful completion of this course, students will be able to:

V2p. Explain how virtual machines, processes, containers and sandboxes work, and the implications for programmers using each.
V2q. Explain process states and state transitions.
V2r. Explain the distinction between supervisor and user permissions, and what that implies for system calls, process management and networking; also explain the role of traps, especially in memory translation.
V2t. Write programs that use interprocess communication, namely pipes and sockets.
V2u. Use simple system calls for common needs.
C2g. Build both ends of a simple producer-consumer link.
C2h. Build a server-side program that uses multi-threading to handle multiple simultaneous clients.
C2i. Identify situations where deadlock may occur, and suggest ways to prevent it.
A2g. 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: Synthesis and Evaluation

Level 3 outcomes are those in which the student can apply the material in new situations. This is the highest level of mastery.

Upon successful completion of this course, students will be able to:

V3w. When a process or a computer is running too slowly, infer some probable causes, including scheduling policy.
C3j. Distinguish when blocking vs nonblocking calls are appropriate.
C3k. Correctly use counting semaphores etc. for queues handling for simple problems.