CS 3432: Computer Organization
Fall 2022
Syllabus

Title:	CS 3432: Computer Organization, CRN 11661 (CRN 15857 for the LAB)
Term:	Fall 2022
Duration:	Tue 08/23/2022 through Thu 12/01/2022
Lecture Time:	Tue, Thur 12:00PM - 1:20PM
Lecture Location:	CCSB 1.0202
Lab Time:	Tue, Thur 10:30AM - 11:50AM
Lab Location:	CCSB 1.0704
Instructor:	Dr. Christoph Lauter cqlauter@utep.edu 915-229-2404
TA:	William Basquez, webasquez@miners.utep.edu, also see course website
IAs:	Ashley Gilmore, amgilmore@miners.utep.edu, Estevan Ramos, ehramos@miners.utep.edu, also see course website
Instructors Office Hours and Location:	Email anytime, Phone during business hours, Office Mon-Thur 2:00PM-3:00PM
Class Web page:	https://www.christoph-lauter.org/utep-arc/
Prerequisites:	

- To succeed in this course, students need familiarity and maturity with concepts and techniques taught in digital design (EE 2369/2169), elementary data structures and algorithms (CS 2401), and discrete math (Math 2300).
- As stipulated in the course catalog, the usual way to demonstrate this familiarity is by earning a C or above in (1) all of these courses and (2) Data Structures (CS 2302).
- Students who earn Bs or above in EE 2369/2169, CS 2401, and Math 2300 and are taking CS 2302 concurrently, are also considered ready.

Lab:

- Students must enroll in the associated lab section.
- Lab sessions and assignments will be managed by the TA.
- Participation is mandatory. It is extremely easy to fall behind and imperative that you make arrangements with the instructor or TA to make-up missed lessons and work. Students at risk of failing due to not engaging with the lab section may be dropped.
- There will be scheduled and unscheduled quizzes in the lab section.

Description:

CS 3432 is a first systems course about how computer systems work. Whereas computer architecture describes the functional behavior of a computer, computer organization describes the structural relationships. CS 3432 is a course in computer organization that answers questions such as the following:

1. How are programs written in a high-level language, such as C or Java, translated into the language of the hardware, and how does the hardware execute the resulting program?
2. What is the interface between the software and the hardware, and how does software instruct the hardware to perform the necessary functions?
3. What determines the performance of a program, and how can hardware designers or programmers improve the performance?
4. What are the reasons for and the consequences of the recent switch from sequential to parallel processing?

Student Learning Outcomes:

See appendix below.

Textbooks:

Required:

- David A. Patterson and John L. Hennessy. Computer Organization and Design RISC-V Edition: The Hardware Software Interface, Second edition, Morgan Kaufmann, ISBN: 978-0128203316.
- Kernighan, Brian, Ritchie, Dennis M. The C Programming Language, Second edition, Prentice Hall, ISBN: 0-13-115817-1.

Optional:

- Other readings and resources will be posted on the course webpage.

Homework Assignments:

- Students are expected to review topics taught in class, work on solutions to assigned problems, and be able to demonstrate skills and solutions during class. Homework assignments will be posted on the course website. Answer will be posted for selected problems.
- To obtain full credit for the class participation and homework grade, you will need to post questions about at least 25 different reading assignments and present solutions to at least 5 homework problems. Questions about reading assignment must be posted before the class for which they are assigned and cannot be made up later.

Workload & Attendance: This course requires participation during the in-class lectures, in addition to the daily readings and the homeworks. Students must read the textbook as well as the posted lecture slides before the corresponding class. Attendance in the lecture will be checked. This course requires student committment to understanding low-level concepts and complex data-structures. Students are encouraged to favor simple and working solutions over complex and incomplete approaches.
Please form study groups. You are encouraged to discuss the approach and understand the problem. However, the write up, programming, and actual solutions to the homeworks are individual work. If you use someone's work for your own, you are committing plagiarism.

Readings: Students are expected to read and understand the textbook chapters before these chapters are discussed in the lecture.

Computers: Students need access to a computer that allows them using a native Linux environment or that allows them `ssh` access to a Linux server, provided by the instructor. For homework assignments that compile in one environment and not the other, the result obtained on the system provided by the instructor will be deemed the canonical result.

Tests and Testing Policy: Three tests will be given: two mid-term tests and a final. All tests are cumulative, with an emphasis on recent material. No make up tests are given but for documented medical emergencies.

Grading:

Letter grade:

Points range	Grade
$[900; 1000]$	A
$[800; 900)$	B
$[700; 800)$	C
$[600; 700)$	D
$[0; 600)$	F

Grading breakdown:

Lab	40%
Homework	15%
Midterm I	10%
Midterm II	10%
Final exam	20%
Attendance	5%

Make-Ups: There will be no make-up for missed exams or homework deadlines, unless due to a documented medical or family emergency. Providing the right kind of documentation for emergencies is the student's responsibility.

Approximate Course Schedule: The course schedule of topics and assignments will be posted on the course website.

Academic Honesty:

- Students are expected to conduct themselves in a professional and courteous manner, as prescribed by the Standards of Conduct: `https://www.utep.edu/hoop/section-2/studentconduct-and-discipline.html`
- Submitted work should be unmistakably your own. You may not transcribe or copy a solution taken from another person, book, or other source (e.g., a web page). Professors are required to report academic dishonesty and any other violation of the Standards of Conduct to the Dean of Students.
- Permitted collaboration: Students may discuss requirements, background information, test sets, solution strategies, and the output of their programs. However, implementations and documentation must be their own creative work. Students are required to document advice received from others and all resources utilized in the preparation of their assignments.
- If academic dishonesty is suspected: The Dean of Students office will be contacted for adjudication. A temporary "incomplete" grade will be issued if their investigation extends beyond the grading period.

Disabilities & Accomodations: If you have a disability and need accommodations, please contact The Center for Accommodations and Support Services (CASS) at 915-747-5148, or by email to `cass@utep.edu`. For additional information, please visit the CASS website at `www.sa.utep.edu/cass`.

COVID19 statement:

- Please stay home if you have been diagnosed with COVID-19 or are experiencing COVID-19 symptoms. If you are feeling unwell, please let me know as soon as possible, so that we can work on appropriate accommodations. If you have tested positive for COVID-19, you are encouraged to report your results to `covidaction@utep.edu`, so that the Dean of Students Office can provide you with support and help with communication with your professors.
- The Center for Disease Control and Prevention recommends that people in areas of substantial or high COVID-19 transmission wear face masks when indoors in groups of people. The best way that Miners can take care of Miners is to get the vaccine. If you still need the vaccine, it is widely available in the El Paso area. For more information about the current infection rates, testing, and vaccinations, please visit `epstrong.org`.

Cell phones: Please silence your mobile devices or put them into a vibrate mode for the duration of class - they are disruptive for your fellow students. No mobile devices (cell phones, PDAs, laptops etc.) are allowed during the exams and will result in your expulsion from the test.

Appendix: CS 3432 Learning Outcomes

Level 1:

- A1. Define and explain the purpose of an instruction set architecture (ISA).
- HSI2. Explain the relationship and differences between a high-level programming language, assembly language, and machine language.
- HSI3. Describe the fetch-execute cycle in terms of the hardware-software interface between machine instructions and processor components.
- NR1. Explain the relationship between a high-level language basic data type (e.g., signed or unsigned integer, floating point number) and its representation as a bit pattern inside the computer.
- A2. Describe the basic components of a processor (e.g., register file, special-purpose registers, control unit, memory) and how they interact with one another.
- HSI4. Explain how procedures are supported by processor hardware.

- HSI5. Explain exception/interrupt handling in terms of the hardware-software interface.
- HSI6. Explain various ways an operand can be addressed in an assembly language instruction.
- HSI7. Describe the process of compiling/assembling, linking, loading, and executing a program.

Level 2:

- NR2. Convert between different integer data representations (e.g., decimal, binary, hexadecimal, octal).
- NR3. Interpret the bit representation of a floating-point number.
- NR4. Perform addition and subtraction on two's complement representation of integers.
- NR5. Use bitwise operators to access and manipulate values stored in a subset of bits within a byte or word.
- NR6. Determine range and precision (if applicable) of numbers that can be stored for a given data type and determine whether an integer operation will result in overflow.
- HSI8. Convert between machine and assembly language representations of instructions – i.e., encode and decode instructions.
- HSI9. Trace the datapath through the processor for a given class of instructions (e.g., arithmetic-logical, memory access, conditional branch)
- HSI10. Trace the execution of an assembly language program with procedure calls in terms of allocation and deallocation of stack frames.
- L1. Translate expressions and assignment statements from C to assembly language.
- L2. Translate Boolean logic and control flow constructs (decisions, loops) from C to assembly language.
- L3. Translate operations on arrays, structs, and pointers from C to assembly language.

Level 3:

- HSI11. Implement/debug simple imperative programs in assembly/machine language.
- HSI12. Write or call a procedure with local variables, parameters, and return value in assembly language.
- HSI13. Implement a simple interrupt handler.
- T1. Compose, compile/assemble, execute, and debug simple programs in a command-line environment, using appropriate modularization and multiple files.