CS333: Storage Systems

Syllabus for Spring 2020 - REVISED FOR REMOTE LEARNING

General Info

Instructor:	Bill Jannen
Email:	jannen@cs.williams.edu
Course Web Page:	`http://www.cs.williams.edu/~jannen/teaching/s20/cs333`
Department Remote Resources:	`http://www.cs.williams.edu/system`

Texts

We will be using the following textbook in this course:

- *Operating Systems: Three Easy Pieces* by Remzi H. Arpaci-Dusseau and Andrea C. Arpaci-Dusseau (version 1.0).

Additionally, a good C reference is highly recommended. Many copies of the following textbook are on reserve at the Schow Science Library:

- *The C Programming Language, 2nd edition by Brian Kernighan and Dennis Ritchie.* by Brian Kernighan and Dennis Ritchie.

Additional readings will be assigned from various sources, including conference proceedings, research journals, magazines, and textbook excerpts. You may also be required to search out and select resources on your own. Additional readings will be digitally accessible from the course website; they will be accessible while connected to the Williams network (or while off-campus using the library's proxy server).

Course Description

This course will examine topics in the design, implementation, and evaluation of storage systems. Topics include the memory hierarchy; ways that data is organized (both logically and physically); hardware characteristics and the ways that storage hardware influences storage software design; data structures and performance models; and system measurement/evaluation. The course emphasis will be on identifying and evaluating design trade-offs.

Course Objectives

Upon the completion of this course, students:

- should have the ability to speak fluently about current and emerging storage technologies;
- should, when presented with a new storage technology, be able to describe how that technology will fit into or modify the "storage landscape";
- should be able to critically read texts in the storage literature and understand the texts' content and arguments;
- should be able to describe and/or use common storage interfaces at the application, OS, and device layers;
- should be comfortable writing system code in C; and
- should be able to analyze the performance of external memory algorithms and data structures.

Course Structure

Course meetings. There will be two recorded units each week. Each set of recordings are associated with one or more required readings. For many units, optional readings are posted to give more variety or depth. These readings are truly optional, but they are encouraged.

Reading Assignments. There will be one or more required readings for each unit. There will be an introductory recording to give context for the material, and I encourage you to watch that recording first. Many of the readings are research papers, so there will be some assumed knowledge and "jargon" that we must sift through before we can understand the papers' core content. The recordings should help with that.

Labs. Lab assignments will continue be an important part of this course. Lab work will be evaluated based on completeness, correctness, and creativity.

Labs may be approached collaboratively, but code must be written by individual groups. Here is what that means:

- Groups may share design ideas with other groups *without limit*.
- Groups may discuss their code with other groups within the following limits:
  - Pseudocode can be shared without restriction.
  - Any and all "real" code written by classmates can be live-reviewed in **READ-ONLY fashion**. Any edits to your group's code must be made by members of your group, but those edits may be directly influenced by discussions with classmates.
  - "Real" code may not be posted or distributed in any persistent format. Source code may be discussed via collaborative editing tools like Atom Teletype. Code may also be discussed using screen-sharing tools like Zoom or Google meet, but these code reviews must be performed in a discussion format.
- Groups may debug with other groups within the following limits:
  - Error messages, edge cases, API behaviors and return values, etc. may be posted and discussed on Piazza.
  - Screen-sharing software may be used to collaboratively debug *in real time*.
  - However, you may not simply ask a classmate to find or fix your error for you.

The overarching rule is that you MAY NOT simply copy code. If you adapt code from another group (which is encouraged), you must recreate it yourself based on your understanding of the ideas. As a rule, all group members are responsible for understanding all aspects of the code you submit.

Note that these policies represent a significant change from previous course guidelines. These changes are designed to adapt to our new "remote learning" reality. Real science should be collaborative; I want us to embrace that spirit with the goal of learning and growing our abilities as systems builders. If you have a question about how any aspect of these rules apply to the honor code, *please ask*. When deciding how to approach any of the course assignments, please ask yourself: "Does this help me to actually learn the material?" If the answer is yes, it is likely allowed.

Exams. We will still have a midterm exam, and it will still be a 24-hour open-book exam. The midterm will be administered on GLOW. You will have a week during which you can choose any time that you would like.

Final Project. For the final assignment of the course, you will either design your own project or select/customize a project proposal from a sample set that I will provide. In the first phase of your project, you will be required to submit a formal proposal that includes success criteria, methods for evaluation, and project deliverables. We will agree upon this proposal together. You will then complete your project, and submit your final writeup/artifact in place of a final exam.

Preparation and participation. We have completed several "hand-in" questions so far this semester. They will be used to calculate your participation grade for the first portion of the semester.

For the "remote learning" portion of the semester, you will be required to complete GLOW activities that correspond to each unit. These will not be graded on correctness. They can be discussed collaboratively without limit. I encourage you to complete the activities in groups using video-conferencing tools. However, you may not post or distribute written solutions.

Grading. Grades for this course will continue to be calculated as follows:

Preparation and participation:	15%
Lab assignments:	35%
Final Project:	25%
Exams:	25%

Note: assignments within a given category may have different weights.

Since we are now reporting course grades on a pass/fail scale, my hope is that we can now focus on learning the material rather than raw scores.

Honor Code

Collaboration is strongly encouraged within the bounds of the guidelines above. However, violations of these new collaboration rules will still be considered a violation of the honor code and will be forwarded to the honor committee. If in doubt of what is appropriate, do not hesitate to ask your instructor. For a full description of the Computer Science Honor Code, please see
`https://csci.williams.edu/the-cs-honor-code-and-computer-usage-policy/`.

Workload.

At Williams, we operate under the course unit system (rather than the credit hour system). You should expect to spend (on average) at least 13 hours per week on the academic and creative work related to class. This includes time spent meeting as a class and working on assignments. The Office of the Registrar explains the relationship of course units to credit hours in greater detail.

Course Calendar and Schedule.

The course calendar and schedule can be found on the course webpage, including links to the readings that should be completed prior to each course meeting (all required textbook materials are available as free PDF documents). The course webpage will be regularly updated after each course meeting to make all lecture materials and examples available for review. Links to lectures will also be posted to GLOW.