CSE113: Introduction to Parallel and Concurrent Programming

January 9, 2023

https://techwireasia.com/2020/06/japans-fugaku-is-the-worlds-fastest-supercomputer/
https://www.lenovo.com/
https://www.apple.com

Hello!

- Professor Tyler Sorensen (he/him)
  - Call me Tyler!
- **Faculty** at UC Santa Cruz Since Summer 2020
  - Third time teaching this class!
- Previously
  - Post doc at Princeton
  - PhD Student at Imperial College London
  - BS/MS at University of Utah

https://users.soe.ucsc.edu/~tsorensen/

https://traveleering.com/
https://wikipedia.com
https://Princeton.edu

Research Interests

MS: Utah

PhD: London

Post Doc: Princeton

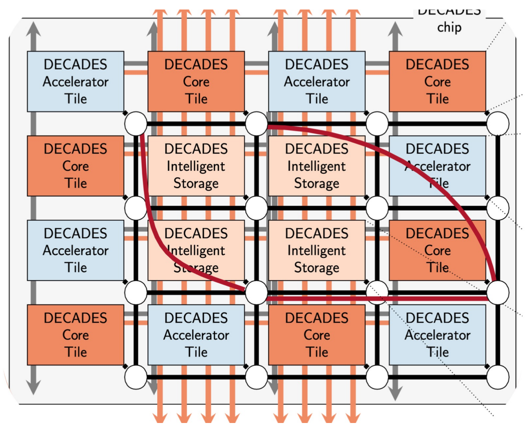
Bugs in GPU programs

locks

barriers

on

new parallel architectures



Research Interests

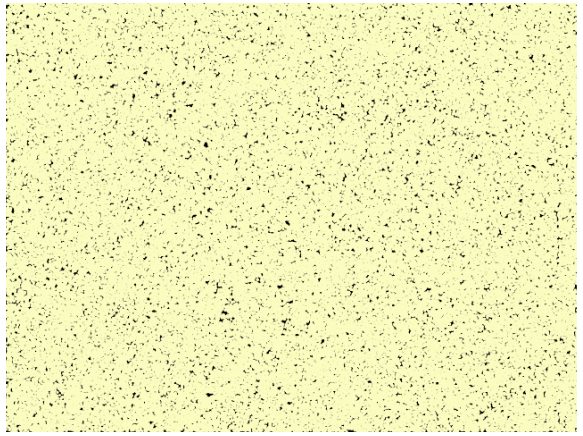
Faculty at UCSC

individual Contributor to

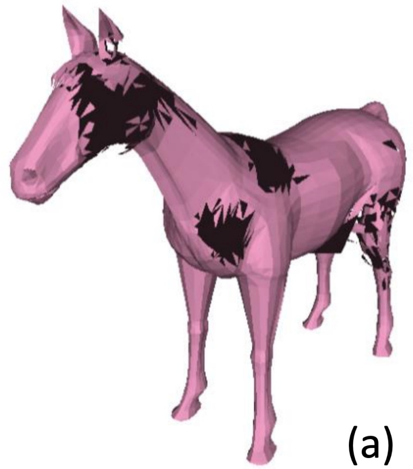
KHRONOS GROUP

Vulkan

parallel particle simulations



GPU memory model testing



Research Interests

- Parallel Programming!

Concurrency and Parallelism is everywhere!

Fujitsu SC at Riken (Japan)

7.6M cores

Consumer Laptop

2-16 cores

Mobile Phone

2-8 cores

Watches?

1 core

BUT

still need to worry about concurrency!

https://techwireasia.com/2020/06/japans-fugaku-is-the-worlds-fastest-supercomputer/
https://www.lenovo.com/
https://www.apple.com

Concurrency and Parallelism is everywhere!

In many cases you won't know what hardware you are programming for

web apps

Android apps

You still need to worry about concurrency!

https://techwireasia.com/2020/06/japans-fugaku-is-the-worlds-fastest-supercomputer/
https://www.lenovo.com/
https://www.apple.com

Learning Objectives

- Foundations of concurrent/parallel computing
  - **Concepts**, not languages/frameworks!
  - Allows you to pick up future new languages and frameworks quickly

- Shared memory concurrency
  - Many concepts apply to other domains, but likely have different performance characteristics (e.g. distributed systems)
  - **Thread cooperation,** not embarrassingly parallel applications

- Think in concurrency
  - understand common synchronization idioms and their performance characterizations
  - efficiently (and safely) utilize modern systems

Wall of text warning for today’s lecture!

- Important to go over class organization and structure
    - You are responsible for knowing this information!
    - All information is also on the class website
    - If you have logistic questions, please refer to the website (or this lecture) first

- Future lectures will be more visual

Today’s Schedule

- Class Structure
- Class Contents
- Assignments, Tests, Grades
  - Special note on new AI tools

Déjà vu....

- Starting the quarter off remote again 😅
  - Current guidance is that we will be remote for today and possibly tomorrow

- We will follow campus guidance on class format, but it seems likely that it will go back to in-person in time for next lecture

- This is not normal!
  - living conditions may have changed
  - uncertainty going forward
  - unrelenting disheartening news

We will persevere!

- We will be:
  - We will be understanding and accommodating with each other
  - We will be kind to each other
  - We will be organized and communicate with each other
  - We will adapt
- We will have:
  - a fun and productive quarter!

Class size

- This is the largest that the class has ever been: 120 students!
- We have lots of help:
  - 3 grad TAs
  - 3 undergrad graders/tutors
- We will need lots of organization and structure to make it through the quarter smoothly.
- We are developing a new class materials, structure, and frameworks to aid with scaling. I appreciate your understanding and patience if there are some growing pains

Teaching Staff Introductions

- Grad TAs:
  - Reese Levine
    - PhD student working on GPU programming models with me
    - 3rd time TA’ing this class
  - Jessica Dagostini
    - PhD student working with me and Prof. Beamer on HPC graph algorithms
  - Devon McKee
    - MS student working with me on GPU synchronization

Teaching Staff Introductions

- Undergrad tutors and graders
  - Sanya Srivastava
    - 2nd time tutoring this class
  - Kyle Little
    - working on a WebGPU project
  - Anish Pahilajani
    - Top student in class last year

Teaching Staff Introductions

- They are all awesome! And they are passionate about parallel programming! Please utilize their office hours and tutoring as much as possible!

- They will be your primary point of contact for technical help throughout the quarter

Class Resources

- **Public**: https://sorensenucsc.github.io/CSE113-wi2023/index.html
  - Slides, schedule, resources
  - There are probably typos: please let me know!
- **Private**: Canvas
  - Homeworks, grades, exams, announcements, recorded lectures, zoom links
- **Private Class forum:**
  - Piazza - invite link incoming

Background

- **CSE 12: systems**
    - C/++ programming
    - Compilation
    - Basic unix command line
- **CSE 101: data-structures and algorithms**
    - Data structure specifications (Queues and Stacks)
    - Reasoning about algorithms (Space and time complexity)
- **CSE 120: architecture (recommended)**
    - Caches
    - Assembly Language basics
    - We will review the basics next lecture

If you do not have architecture, please consult the architecture reference on webpage!

Class Format

- **9:20 - 10:25 MWF: 65 minutes**
  - I will try to be 10 minutes early and stay 10 minutes afterwards
  - Class is generally structured as follows:
    - Announcements (homework assignments, etc.)
    - Quiz review
    - Previous lecture review
    - New material

- Please be engaged and participate in class! questions, comments, corrections, etc. are all welcome!

Class format

- Once things return to in-person (hopefully on Wednesday🤞)

- This is an in-person synchronous class
  - I expect you to make an effort to attend the synchronous lecture
  - Please participate in class and network with your classmates

Class format

- But don’t come to class sick!
- I plan to record lectures and release through Yuja ASAP.
- Watch the lecture and then do the associated quiz.

Class format

- This class is designed to be a synchronous in-person class. Lecture recordings are not meant to be an equal substitute for live class. They are meant to fill in occasionally.

- If you do not plan to attend the majority of the classes in-person, please talk to advising to find a class better suited for you.
  - Inevitably, students that do not attend lectures do not do as well in the class

- If synchronous attendance drops significantly, then we will revisit attendance grading and lecture recordings.

Office Hours

- **My office hours:**
  - 3 - 5 PM on Thursdays
  - I will share a google sign up sheet (it will contain a zoom link)
  - Slots are 10 minutes
  - link will be posted in Canvas around noon that day
    - don’t sign up before the Canvas announcement
    - don’t sign up unless you have a question
    - sign up for 1 slot at a time
  - Strict with timing to make sure it is fair

Office Hours

- The TAs and tutors will organize their office hours this week and we will let you know the days/times
  - We will strive to get a mix throughout the week and have both hybrid and in-person

Asynchronous Discussion

- **Piazza**
  - Private message (to teaching staff) technical homework questions, sensitive questions
  - Programming and framework questions (global)
  - Tech news (global)
  - Discussions on class material (global)

- Please do not email me personally unless it is extremely sensitive.

- Do not expect replies off-hours (after 5 pm, weekends, holidays)

We will try to answer in 24 hours
Please try to help your peers!

Asynchronous Discussion

- **Additional forums**
  - You are welcome to create one yourselves
  - Please make it open and available to all your classmates
  - Please provide sufficient moderation (e.g. be nice to each other!)
  - Please do not cheat
  - Please remember that anything that is not in Canvas may not be private

  - If there are issues, please let me or a TA know!

Class Content

Class Content

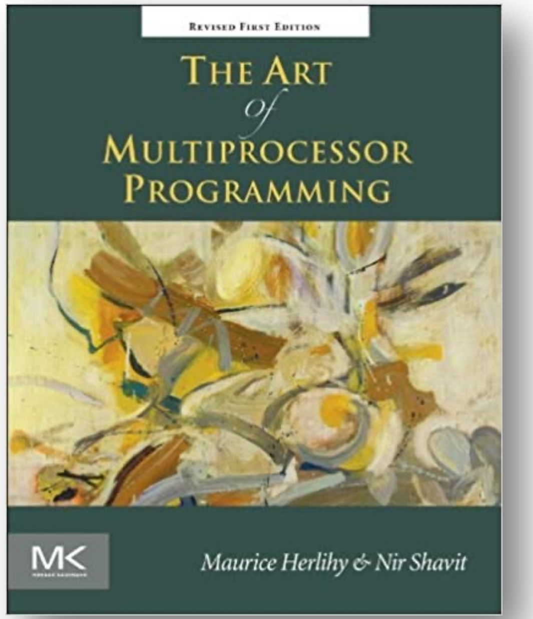
- **30** classes, split into
- **5** modules, so there are
- **6** classes per module

- **Reference book:**

Available online from the library

Link on the webpage

Book uses Java: we will use C++


REVISED FIRST EDITION
THE ART of MULTIPROCESSOR PROGRAMMING
Maurice Herlihy & Nir Shavit
MK

Class Content

- **Module 1: Introduction, Background and ILP** This module will provide an architectural refresher and discuss how modern hardware exploits parallelism within a thread (ILP). We will also introduce threading in C++.

Class Content

- **Module 2: Mutual Exclusion** This module will discuss the fundamental problem of mutual exclusion. We will discuss the theory behind mutual exclusion, how it is implemented in practice, and specialized mutual exclusion objects.

Class Content

- **Module 3: Concurrent Data Structures** This module will discuss concurrent objects and how to reason about them. We will discuss several implementations and discuss how it can be used in load balancing and software pipelining.

Class Content

- **Module 4: Reasoning about Concurrency** This module will discuss how to reason about concurrent programs, including memory consistency and fairness.

Class Content

- **Module 5: Heterogenous Parallelism (GPGPU)** This module will discuss heterogenous programming, with a focus on GPGPU programming. We will discuss the SIMT programming model, hierarchical execution, and different architectural considerations when optimizing programs.

- *I’m very excited for this module! We want to use a new platform (WebGPU) to allow everyone to program the GPUs on their own machines!*

Class Content

- **Schedule:**

https://sorensenucsc.github.io/CSE113-wi2023/schedule.html

Readings are highly recommended will be a useful reference for test studying and homeworks

Slides will be uploaded before lecture

Accessibility

UC Santa Cruz is committed to creating an academic environment that supports its diverse student body. If you are a student with a disability who requires accommodations to achieve equal access in this course, please submit your Accommodation Authorization Letter from the Disability Resource Center (DRC) to me by email, preferably within the first two weeks of the quarter. I would also like us to discuss ways we can ensure your full participation in the course. I encourage all students who may benefit from learning more about DRC services to contact DRC by phone at 831-459-2089 or by email at drc@ucsc.edu.

Assignments and Tests

Assignments and Tests

- **Assignments**:
  - 1 assignment per module
  - halfway through the module
  - due in 10 days
  - Each homework is worth 10% of your grade (total of 50%)

Do not expect replies off-hours (after 5 pm, weekends, holidays)

- We will try to make homeworks due at midnight. If we receive too many questions off hours, we will move earlier (e.g., 8 PM)

Assignments and Tests

- **Format**:
  - Coding assignments in C/++ and Python (and some Javascript for module 5)
  - We will provide a docker image that you should be able to run locally.
- *It is recommended that you have access to a machine with at least 4 cores!*
  - If you do not, then our new format should be sufficient for you.

Assignments and Tests

Homework framework redesign planned (led by TAs).

We aim to use github classroom for submission and automatic feedback.

You will be graded on the server feedback rather than the results from your own machine. This is to help provide fair (and scalable) grading across the increasing diversity of devices that everyone has these days. Someone with an M1 processor will get very different results than someone with an Intel X86 processor.

There may be some friction getting started. We appreciate your patience and understanding. I will update the class as we make progress.

Ultimately this should be a big quality-of-life improvement for you!

Assignments and Tests

Architectural differences are very interesting to discuss and I hope we can have detailed discussions about how your machine's results differ from the server on Piazza

Assignments and Tests

Two tests: Final and Midterm

- We did asynchronous tests last year. Let’s plan on that again this year.
- Designed to take ~180 minutes
- As a warning: people take much longer on take-home tests than in-person tests!

Assignments and Tests

Midterm

- Assigned halfway through module 3
- One week
- Assigned Feb 13, Due Feb 17
- Designed to take 3 hours
- Worth 10% of grade
- *Review slides and readings*

Do not expect replies off-hours (after 5 pm, weekends, holidays)

Assignments and Tests

Final

- 1 Day (12 hours)
- Assigned 8 AM on March 22
- Due 8 PM on March 22
- Designed to take 3 hours, we will monitor Piazza
- Scheduled time is 4 PM to 7 PM (when we will be most active with help)
- Worth 30% of grade
- *inclusive: slide material from all year, including readings*

Assignments and Tests

What you can use for tests:

You are free to consult notes, books, or the internet. While the test is active, you are not allowed to discuss the test with another person (either in the class or online). For example, you *can* google concepts that are on the test. You *cannot* post a test question to stackoverflow.

You are not allowed to use AI tools (github co-pilot or chatGPT). That will be considered cheating.

Please do not cheat! I like asynchronous tests but if we catch students cheating then I will have to move to synchronous tests!

Assignments and Tests

For assignments:

I expect submitted assignments to contain your own original work. You can refer to notes, slides, internet, etc. But do not blindly copy code.

Any part of your submission that is not your original work (e.g. code snippets from the internet) need a citation. My aim is to be lenient with cited code, but we may remove some points based on the extent. A few missing points is better than a referral for academic misconduct.

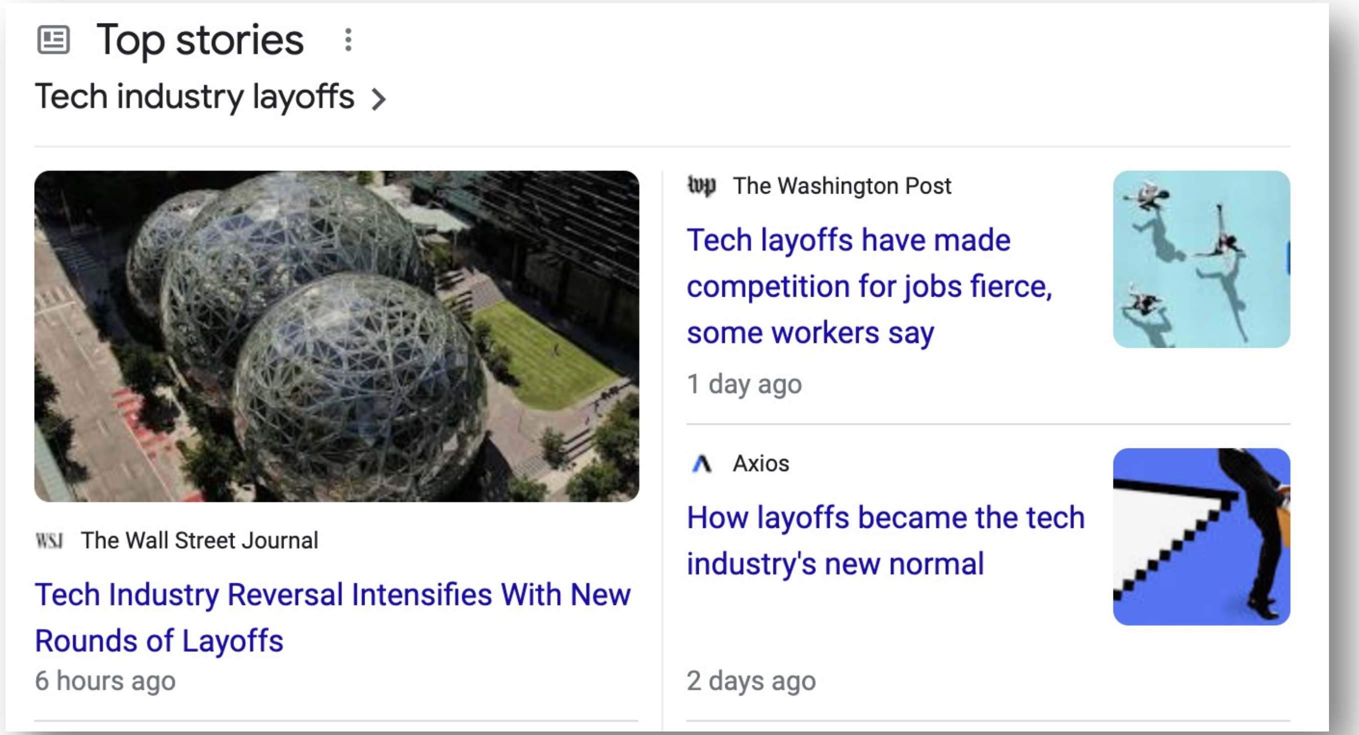
I prefer that you do not collaborate on homework with classmates. In the case that you do, please mention in the submission. Again, a few missing points is better than a misconduct referral.

Cheating

This class has a zero tolerance policy on cheating. Please don’t do it. I would much rather get a hundred emails asking for help than have to refer anyone for academic misconduct.

Cheating harms you: this is the best chance in your career to take the time to really learn the class material. If you do not learn the material you will not be successful in a tech career.

The current economic conditions are volatile for computer science graduates. You will not stand out to a company for having straight As. You will stand out if you can show a deep understanding of complicated CS topics. When you cheat, you deprive yourself of this learning.


Top stories

Tech industry layoffs >

The Wall Street Journal

Tech Industry Reversal Intensifies With New Rounds of Layoffs

6 hours ago

The Washington Post

Tech layoffs have made competition for jobs fierce, some workers say

1 day ago

Axios

How layoffs became the tech industry's new normal

2 days ago

Discussing results

You cannot share code snippets or discuss coding solutions at a low-level.

However, unless otherwise specified: in the second week of the assignment, you can share local (from your own machine) results with your classmates.

You will have different machines and thus, your results may not align completely: it is interesting to think about why!

AI Tools

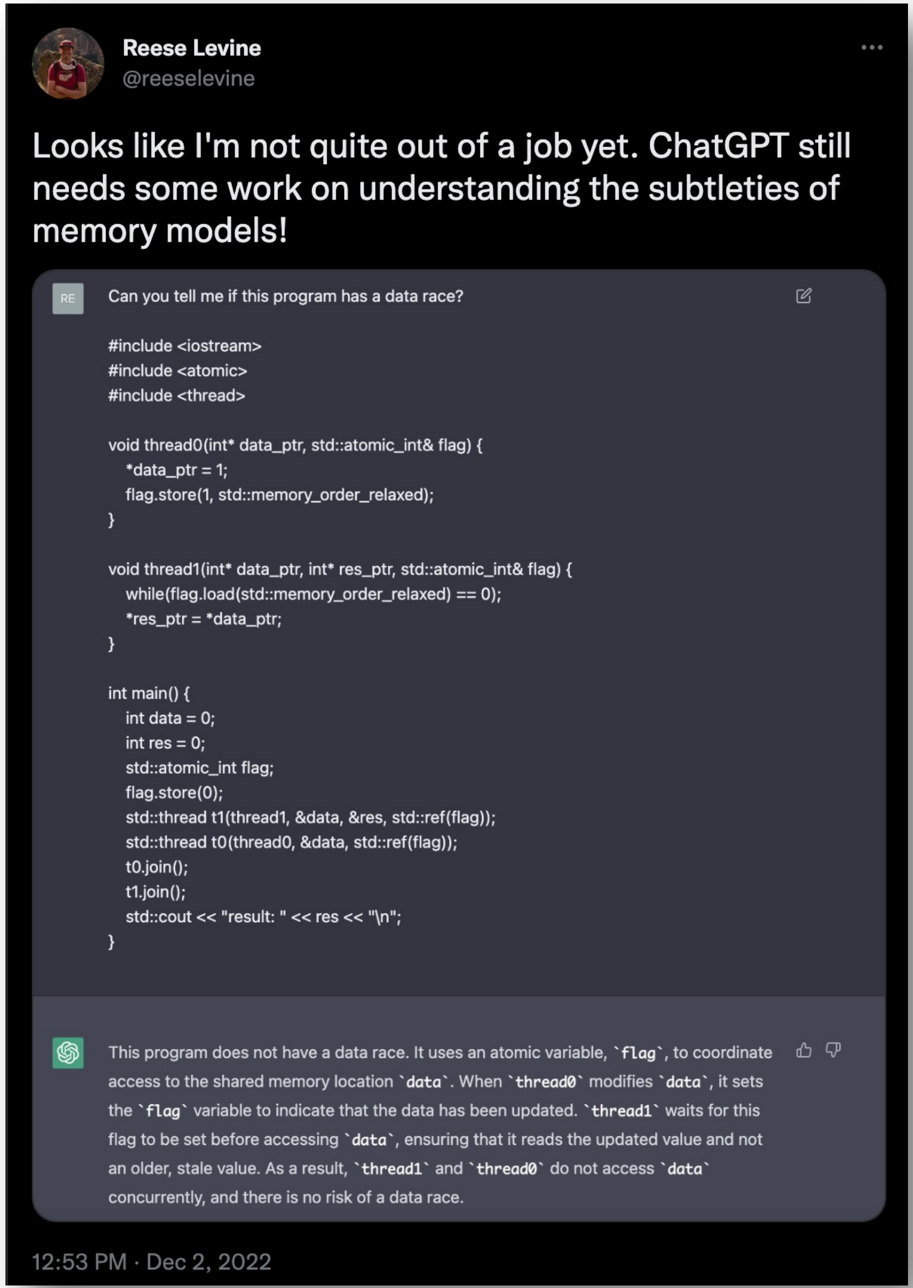
- Exciting time for AI:
  - Github co-pilot
  - ChatGPT
- Impact on learning objectives is not clear.
- This class has been designed to be taken *without* the use of AI tools.
  - If you use them for your assignments (or especially tests), it will be considered academic misconduct. To be clear: it is not acceptable to use the tools and modify the AI output. Just do not use the tools at all when doing coursework for this class.

AI Tools

- However, I will provide the following option to you (only for the homeworks):
- *after* you have completed the assignment, you can use the AI tools to see how they could help solve the problem.
  - Please submit a non-AI version of the assignment first.
  - Afterwards, try to use the AI tools. Write several paragraphs explaining how you approached the tools, what prompts you used, what the tools suggested, and if the tools were correct or incorrect.

- Submitting this extra report can provide up to 5 points extra credit (not to exceed 100%) on each homework assignment. There is no rubric for the extra credit and points will be given at our discretion.

AI Tools



Reese Levine
@reeselevine

Looks like I'm not quite out of a job yet. ChatGPT still needs some work on understanding the subtleties of memory models!

Can you tell me if this program has a data race?

```
#include <iostream>
#include <atomic>
#include <thread>

void thread0(int* data_ptr, std::atomic_int& flag) {
  *data_ptr = 1;
  flag.store(1, std::memory_order_relaxed);
}

void thread1(int* data_ptr, int* res_ptr, std::atomic_int& flag) {
  while(flag.load(std::memory_order_relaxed) == 0);
  *res_ptr = *data_ptr;
}

int main() {
  int data = 0;
  int res = 0;
  std::atomic_int flag;
  flag.store(0);
  std::thread t1(thread1, &data, &res, std::ref(flag));
  std::thread t0(thread0, &data, std::ref(flag));
  t0.join();
  t1.join();
  std::cout << "result: " << res << "\n";
}
```

This program does not have a data race. It uses an atomic variable, `flag`, to coordinate access to the shared memory location `data`. When `thread0` modifies `data`, it sets the `flag` variable to indicate that the data has been updated. `thread1` waits for this flag to be set before accessing `data`, ensuring that it reads the updated value and not an older, stale value. As a result, `thread1` and `thread0` do not access `data` concurrently, and there is no risk of a data race.

12:53 PM · Dec 2, 2022

Late policy

- Assignments:
    - You have 10 days to submit each assignment.
    - Each assignment has 4 days that you can turn in the assignment late with no penalty.
    - No work accepted after the 4 days.

- Tests:
    - Will not be accepted late. No exceptions

Reviewing Grades

- For assignments and tests:
  - You have 1 week from when the grade is posted to discuss grades with teaching staff

Assignments and Tests

Grade Breakdown:

- 5 homeworks: 50%
- 1 midterm: 10%
- 1 final: 30%
- **attendance/quiz**: 10%

Attendance and Quizzes

- Small canvas “quiz” every lecture - take the quiz to get the daily points
- Quiz answers are not graded! only if you submit it
  - However, low-effort quiz submissions are liable to be failed.
- Quizzes are posted after class and due before the next class
- Some quiz questions do not have a right or wrong answer. They are meant to make you think about the material!

Attendance and Quizzes

- You can miss up to 3 quizzes without penalty.

- Only submit the quiz once you have watched the lecture (either in person or remotely)! if you do not watch the lecture and submit the quiz, it is considered a breach of academic integrity.

Website tour

Final notes

- This class is still “new”
  - Material is still being developed, especially at this scale
  - There may be issues on HWs and tests (please let us know if you find any!)
  - There may be schedule changes

We will do our best and make sure to stay organized and communicate clearly!

Thank you!

- It’s a risk taking a new class like this. Thank you for giving it a chance!
- Your experiences and feedback will help shape this class for future students.
- Email is always open for comments about class material, HW assignments, etc.

Next Class

- **Architecture/Compiler review:**
  - Why?
  - Parallel programming lives at the edge of the software/hardware interface. We will need to understand architecture/compiler basics in order to program efficient and correct programs

  - *Good programming languages for parallel architectures is still an open problem!!*