15-410

"My other car is a cdr" -- Unknown

Exam #1
Oct. 28, 2020

Dave Eckhardt

Dave O'Hallaron

Synchronization

Checkpoint 2

- Monday during class time
- Again, you will receive a time slot and Zoom coordinates
  - We expect everybody can make every time slot
  - If you have a conflict, inform us *by Saturday evening*
- Your kernel should be in mygroup/p3ck2

Checkpoint 2 - alerts

- Reminder: context switch ≠ timer interrupt!
  - Timer interrupt is a *special case*
  - Looking ahead to the general case can help you later
- Please read the handout warnings about context switch and mode switch and IRET *very carefully*
  - Each warning is there because of a big mistake which was very painful for previous students

Synchronization

Book report!

- This your approximately-mid-semester reminder about the book report assignment

Synchronization

Asking for trouble?

- If you aren't using source control, that is probably a mistake
- If your code isn't in your 410 AFS space every day, you are asking for trouble
  - GitHub sometimes goes down!
    - » S'13: on P4 hand-in day (really!)
  - Roughly 1/2 of groups have blank REPOSITORY directories...
- If your code isn't built and tested on Andrew Linux every two or three days, you are asking for trouble
  - Don't forget about CC=clang / CC=clangalyzer
- Running your code on the crash box may be useful
  - But if you aren't doing it fairly regularly, the first "release" may take a *long* time

Synchronization

Google "Summer of Code"

- http://code.google.com/soc/
- Hack on an open-source project
  - And get paid
  - And quite possibly get recruited
- Projects with CMU connections: Plan 9, OpenAFS (see me)

CMU SCS "Coding in the Summer"?

Synchronization

Debugging advice

- Once as I was buying lunch I received a fortune

Synchronization

Debugging advice

- Once as I was buying lunch I received a fortune

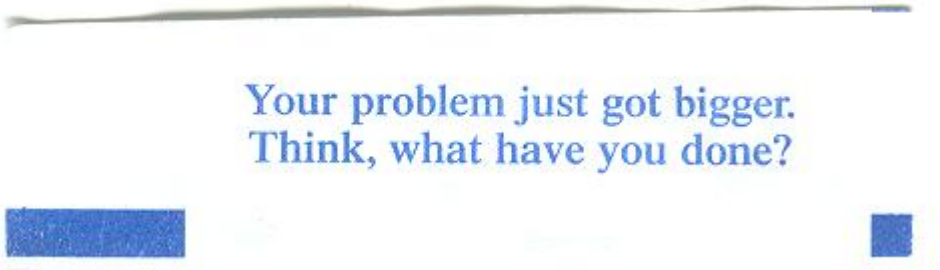


Image credit: Kartik Subramanian

A Note for Posterity

The F'20 mid-term exam occurred during COVID-19

This was an atypical exam

- "2 hours of material"
- 4-hour exam session
- Personal start time in a 36-hour window
- Open book, open notes (including submitted P0/P1/P2 code)
  - Honor system
- Reduced weight at the end of the semester

A Word on the Final Exam

Disclaimer

- Past performance is not a guarantee of future results

The course will change

- Up to now: “basics” - What you need for Project 3
- Coming: advanced topics
  - Design issues
  - Things you won't experience via implementation

What will that mean for the final exam?

- We don't know yet!
- Early advice
  - “Attend” lectures, do readings
  - Review your code and your partner's code
  - Review ink comments from the course staff

“See Course Staff”

If your exam says “see course staff”...

- **...you should!**

This generally indicates a serious misconception...

- **...which we fear will seriously harm code you are writing now...**
- **...which we believe requires personal counseling, not just a brief note, to clear up.**

...though it might instead indicate a complex subtlety...

- **...which we believe will benefit from personal counseling, not just a brief note, to clear up.**

“See Instructor”...

- **...means it is probably a good idea to see an instructor...**
- **...it does not imply disaster.**

“Low Exam-Score Syndrome”

What if my score is really low????

- **It is frequently possible to do *dramatically* better on the final exam**
- **Specific suggestions later**

Outline

Question 2

Question 3

Question 4

Question 5

Q2 – P2 design decision

Purpose: demonstrate grasp of a design tool

- Hopefully P2 involved deliberate design
- Hopefully P3 is involving deliberate design
- "Robust code is *structurally different* than fragile code"
- P3 requires not just code but *structurally non-fragile code*.

If you were lost on this question...

- We had a lecture on this topic (September 4)
- Other "odd" lectures to possibly review
  - Debugging, Errors
  - #define, #include
  - We expect you to know *and apply* all of this material

Q2 – P2 design decision

The chart format is your friend

- Without a chart it is too easy to forget to compare the same factors across all proposals
  - "Pros and cons" faces this danger
  - A feature matrix without metric names plus values *that match the names* faces this danger

Look for third/fourth options

- Conflict between desirable factors in two proposals can inspire a new proposal
- "Pros and cons" hides these conflicts

Q2 – P2 design decision

Use numbers when possible

- Avoid "pseudo-booleans"
  - Avoid: "good performance" with values "yes" and "no"
  - Prefer: "run time" with values "O(N)" and "O(logN)"

Be specific when possible

- Avoid: "freeing of resources"
- Prefer: "freeing of thread control block"

Be wary of "dangerous metrics"

- "Does it work?" / "Fundamental correctness"
  - Documenting non-working proposals can be useful in some situations
  - But two non-working proposals plus one working proposal probably means that design work should continue

Q2 – P2 design decision

Conclusion form

- Avoid
  - We picked X.
  - We picked X because it was the only correct solution.
- Prefer
  - We picked X because value V1 for M1 is unacceptable for the expected workload.
  - We picked X because (M1, V1) is more important than (M2, V2).

Q2 – Overall

Scores

- 70% of the class scored 5/6 (83%) or better

Q3 – Register Dump

Question goal

- Stare at a register dump and form a plausible hypothesis
  - Why? Debugging P3 will require staring at bits to figure out what's wrong... this is a good way to figure out if some practice is needed

Hint

- Something is about to run into something painful

Common issues

- It is necessary to say *why/how* a wrong register leads to an exception
  - Since there was a fault, it ought to be possible to say why some particular instruction failed to execute

Q3 – Overall

Scores

- **~45% of the class scored 4/4**
- **~20% of the class scored 3/4 (75%)**

Q4 – sem_broadcast()

What we were testing

- Find a synchronization botch (important skill)
- Write a convincing trace (demonstrates understanding)

Good news

- ~2/3 scored 16/20 or better

Less-good news

- ~20% scored 12/20 or below
- Equivalent to not finding a second problem at all

Q4 – sem_broadcast()

Minor issues

- Omitting too many lines of trace (e.g., conditional checks)

Noticeable issues

- Not explicitly naming an observed problem
- Not giving a clear and compelling trace
  - A verbal description is usually insufficient

Semantic issues

- Invoking sem_broadcast() on a “binary semaphore” isn't a synchronization problem in the code we provided
  - There were two issues in the provided code, one which ranks as “most grievous”
- There is no rule that semaphores are capped at their initial value

Q5 – Condition variables via rendezvous()

Question goal

- Atypical variant of typical “write a synchronization object” exam question
  - Write condition variables using mutexes and rendezvous()
- This question included an atypical hazard source
  - Namespace collision
    - » Results: false wakeups, maybe threads getting stuck
    - » Unusably severe in certain approaches

Q5 – Condition variables via rendezvous()

Question goal

- Atypical variant of typical "write a synchronization object" exam question
  - Write condition variables using mutexes and rendezvous()
- This question included an atypical hazard source
  - Namespace collision
    - » Results: false wakeups, maybe threads getting stuck
    - » Unusably severe in certain approaches

Frequent approaches

- ~20%: rendezvous() tag is a counter
- ~50%: rendezvous() tag is a tid
- ~20%: rendezvous() tag is an address
  - Some used the "value" parameter too

Q5 – Condition variables via rendezvous()

Other issues

- According to the 15-410 orthodoxy, mutexes should be held *only briefly*
  - "For an indefinite time" does not count as brief!
- Synchronization objects that are LIFO are factories for a bad thing
  - The first letter of the name is 's'
- Don't let go of the "world mutex" too early!
  - This is a small detail, but a serious conceptual issue
  - See course staff if necessary

Time

Our target: "2 hours of exam content"

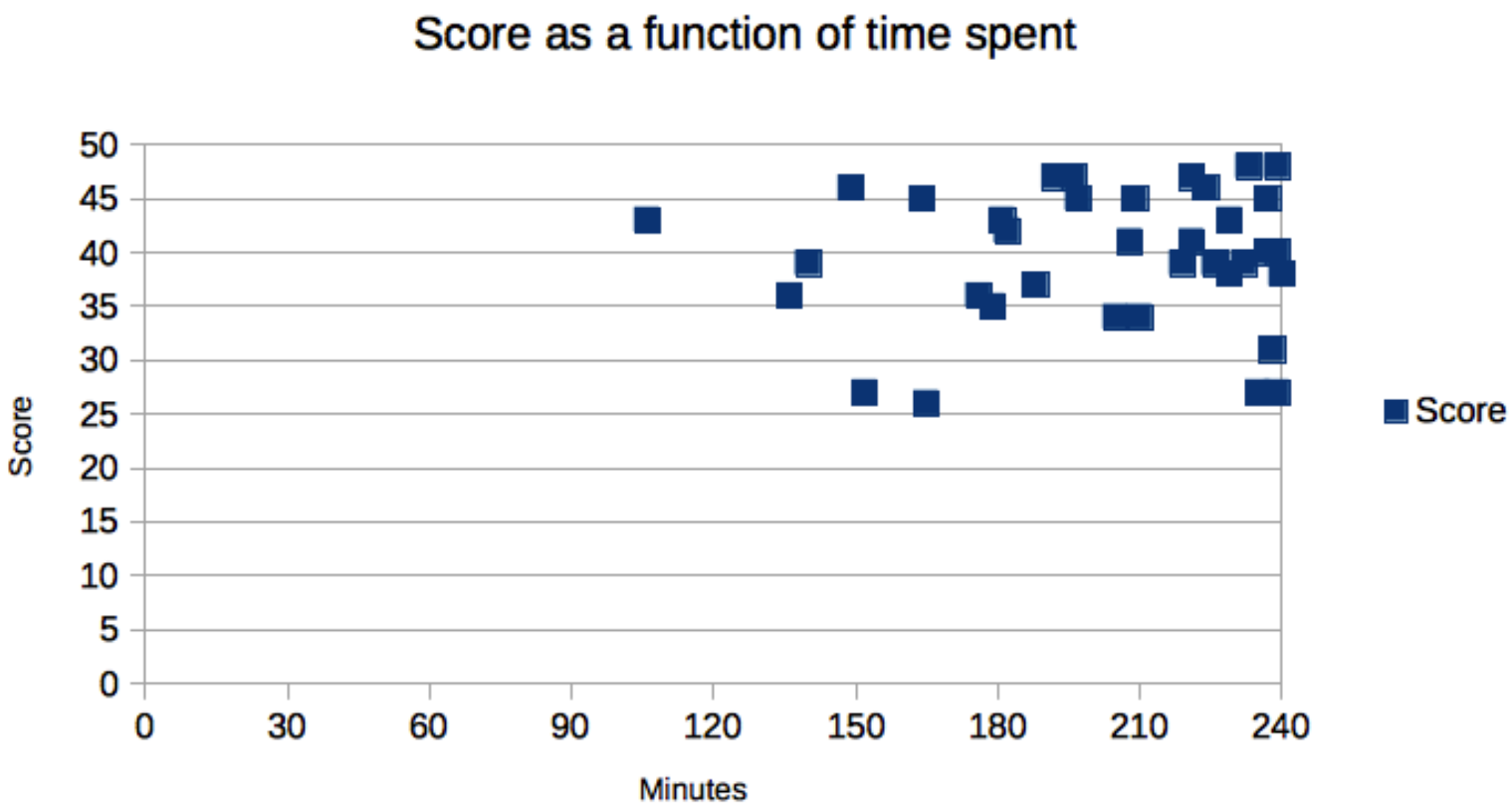
- Provided: 4 hours of exam time

Observations

- Min: 1.75 hours
- Median: 3.5 hours
- Max: 4 hours

Wow, almost nobody took under 2 hours?

Time


Score as a function of time spent
Score
50
45
40
35
30
25
20
15
10
5
0
0
30
60
90
120
150
180
210
240
Minutes
Score

Time

Our target: "2 hours of exam content"

- Provided: 4 hours of exam time

Observations

- Min: 1.75 hours
- Median: 3.5 hours
- Max: 4 hours

Wow, almost nobody took under 2 hours?

- Time after 200 minutes (3:20) doesn't look all that fruitful
- Low-ish scores range from ~2.5 hours to 4 hours
- High-ish scores range from ~2.5 hours to 4 hours

Breakdown

```
90% = 45.0   11 students
80% = 40.0    8 students
70% = 35.0   11 students
60% = 30.0    4 students
50% = 25.0    4 students
40% = 16.0    0 students
<40%          0 students
```

Comparison/calibration

- Scores are high compared to a typical 410 mid-term
  - Low of 52%, median of 79%
- But these are atypical conditions in many ways

Implications

Score below 35?

- Form a “theory of what happened”
  - Not enough textbook time?
  - Not enough reading of partner's code?
  - Lecture examples “read” but not grasped?
  - Sample exams “scanned” but not solved?
- It is important to do better on the final exam
  - Historically, an explicit plan works a lot better than “I'll try harder”
  - *Strong suggestion*:
    - » Identify causes, draft a plan, see instructor

Implications

Score below 30?

- Something went *noticeably* wrong
  - It's *important* to figure out what!
- Passing the final exam could be a challenge
- *Passing the class may be at risk!*
  - To pass the class you must demonstrate proficiency on exams (not just project grades)
  - We don't know the format of the final exam yet, but a strong grasp of key concepts, especially concurrency, is important
- Try to identify causes, draft a plan, see instructor
  - Good news: explicit, actionable plans usually work well

Action plan

Please follow steps in order:

1. Identity causes
2. Draft a plan
3. See instructor

Action plan

Please follow steps in order:

1. Identity causes
2. Draft a plan
3. See instructor

Please avoid:

- "I am worried about my exam, what should I do?"
  - *Each person should do something different!*
  - Thus "identify causes" and "draft a plan" steps are individual and depend on some things not known by us

Action plan

Please follow steps in order:

1. Identity causes
2. Draft a plan
3. See instructor

Please avoid:

- "I am worried about my exam, what should I do?"
  - *Each person should do something different!*
  - Thus "identify causes" and "draft a plan" steps are individual and depend on some things not known by us

General plea

- Please check to see whether there is something we strongly recommend that you have been skipping because you never needed to do that thing before
  - This class is different