

Lecture 9

COMP207

What are we doing today?

- Go over the weekly assessment
- Go over the weekly survey
- Discuss undo-, redo-, undo/redo-logging
- Go over previous exam

Weekly survey

(Skipping the comments mentioned Thursday)

Positive:

- Feedback is clearly taken seriously, which is greatly appreciated!
- The progression of content is easy to follow

Nice! I am happy to hear that!

Weekly survey part 2

Negative:

- I feel like the tutorial videos brushed over special locks maybe a bit quickly for me. I didn't fully understand differences between upgrading vs updating locks, when everything is allowed etc

Response: There is no difference – I should be consistent with the naming but they are the same thing. I will try to show by example what the tables for when things are allowed mean, but most are fairly straightforward if you try to understand what they are trying to say (the exception is Upgrade locks because they are not symmetric, but the example from last time was supposed to help with that)

Weekly survey part 3

- Sometimes can't read English sentences or question stems very clearly

Response: I have heard similar comments previously. When I ask for clarification about what sort of things they are talking about, they typically show questions that are direct, but the student expected there to be a trick to it to make it easy. However, there simply was no trick – it was just a hard question

Weekly survey part 4

Negative:

- 1) It's hard to pay attention in lectures and make notes
- 2) Some examples take longer than needed to explain
- 3) Slides could be laid out clearer
- Response: 1) It is part of why I wanted to do it this way: The definitions and so on where you really need to understand it the given way are on videos and you can rewatch them as needed.
- 2) this is a balance – I can go faster through examples, but some will struggle to follow at that point. It is not helpful for this that there is a wider spread in backgrounds than in most other courses.
- 3) I would need some examples of that and what you had in mind

Weekly survey part 5

Negative:

- Not specific to this week: there is quite a lot of video content to get through each week. In combination with the lectures, labs, revision, etc., the workload for this module feels unusually high
- Response: I will try to cut down on it a bit.
That said, while it might be unusually high, it still seems lower than what it should be: 10 hours per week. 3 hours are for labs and lectures, so 7. At a guess, I think people use around 10-20 hours for the main assignment – let's put it high, which leaves us at 5 hours. I think the videos average to around 1:30 a week, so we are left with 3:30 a week for doing quizzes and other revisions, which seems like a fairly big amount

Undo-, redo-, redo/undo-logging

- We saw 3 kinds of logging
- They write something in a log file when a transaction: Starts (not too important), writes, commits, rollback.
- Besides writes it is straightforward: Just write what transaction did what

Undo-, redo-, redo/undo-logging: writes

- To do Undo logging, we need to be able to undo transactions (in particular the writes) to roll them back, so we include in the log what the content of the variable was before we made the write:
 - $\langle T1, X, 10 \rangle$ if transaction T1 changed X **from** 10 to something else
 - Undo works backwards from the end of the file to the beginning
- To do Redo logging, we need to be able to redo transactions that have committed, so we include in the log what the content of the variable was after the write
 - $\langle T1, X, 10 \rangle$ if transaction T1 changed X **to** 10 from something else
 - Redo works from start to end
- To do Undo/Redo, you first undo uncommitted transactions and then redo committed transactions
 - $\langle T1, X, 10, 20 \rangle$ if transaction T1 changed X from 10 to 20

What is best?

- Doing Undo without Redo requires Force: You must finish writing the changes to disk before you commit
 - It is cheaper to wait until you have many changes and then write them all together so this is expensive
 - Not having Force is called No Force
- Doing Redo without Undo requires No Steal: You must never overwrite committed data with uncommitted data on disk
 - Typically disks are slow but you can make a big change at once so avoiding writing uncommitted data to disk requires you to be careful about this
 - Not having No Steal is called Steal

Go over previous exam