School of Computer Science & Engineering

COMP9242 Advanced Operating Systems (AOS)

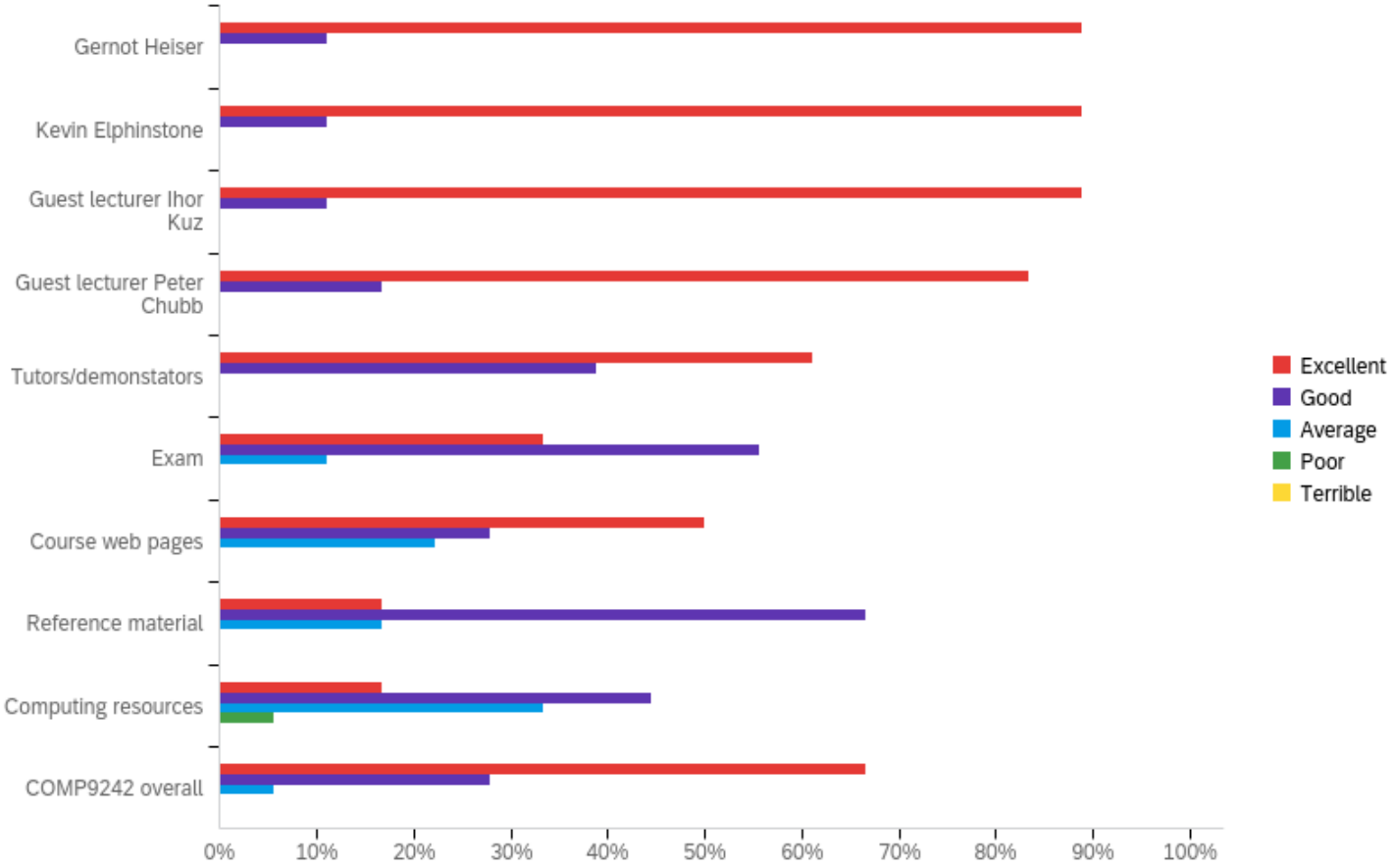
2022 T2

AOS Course Survey Result

@GernotHeiser

Copyright Notice

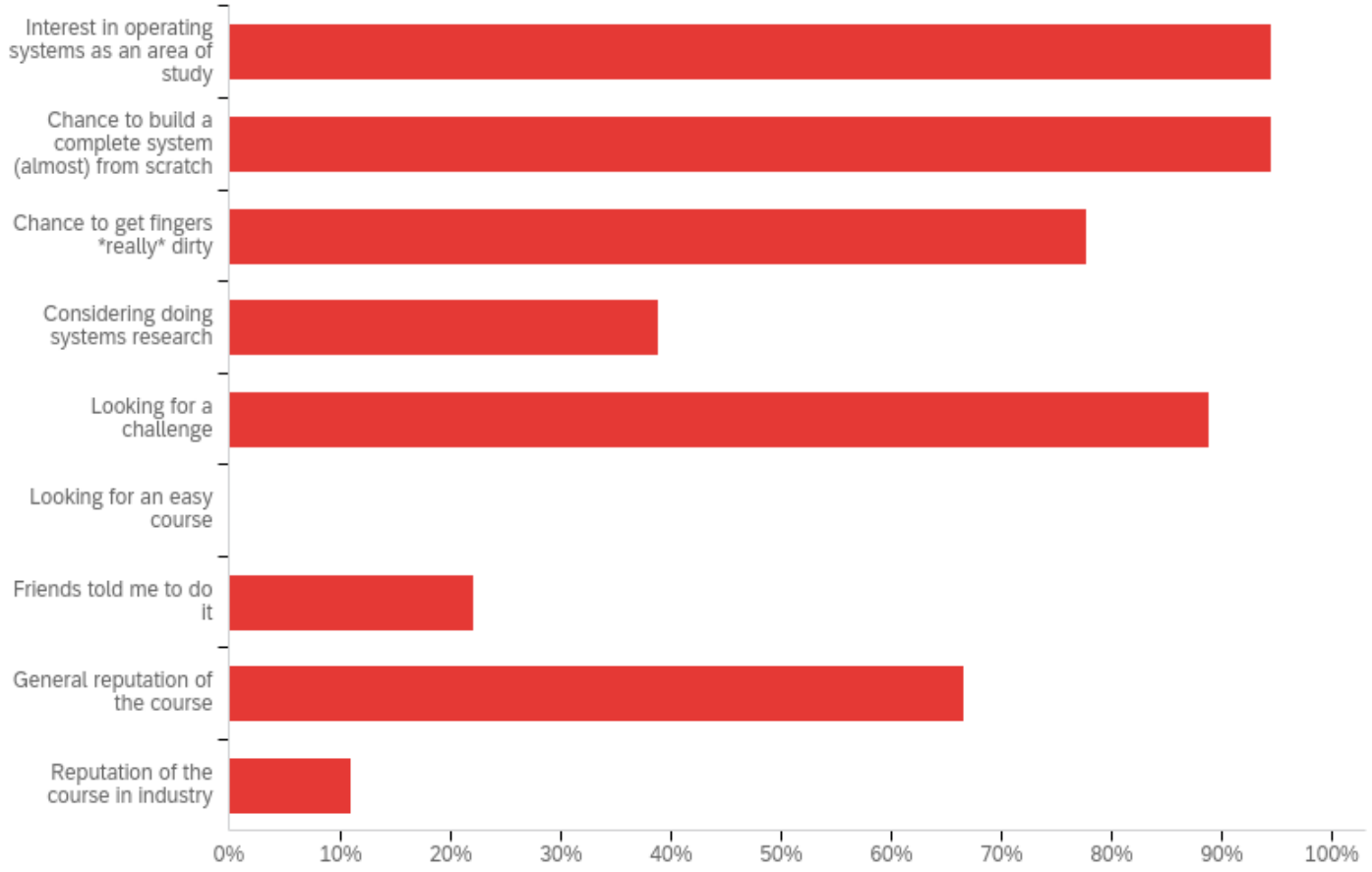
Q1: Quick evaluation single answer



Q1: Quick evaluation single answer

Question	Mean	Excellent		Good		Average		Poor		Terrible		Total
Gernot Heiser	1.11	89%	16	11%	2	0%	0	0%	0	0%	0	18
Kevin Elphinstone	1.11	89%	16	11%	2	0%	0	0%	0	0%	0	18
Ihor Kuz	1.11	89%	16	11%	2	0%	0	0%	0	0%	0	18
Peter Chubb	1.17	83%	15	17%	3	0%	0	0%	0	0%	0	18
Tutors/demonstators	1.39	61%	11	39%	7	0%	0	0%	0	0%	0	18
Exam	1.78	33%	6	56%	10	11%	2	0%	0	0%	0	18
Course web pages	1.72	50%	9	28%	5	22%	4	0%	0	0%	0	18
Reference material	2.00	17%	3	67%	12	17%	3	0%	0	0%	0	18
Computing resources	2.28	17%	3	44%	8	33%	6	6%	1	0%	0	18
COMP9242 overall	1.39	67%	12	28%	5	6%	1	0%	0	0%	0	18

Q2: Your main reasons for taking AOS?



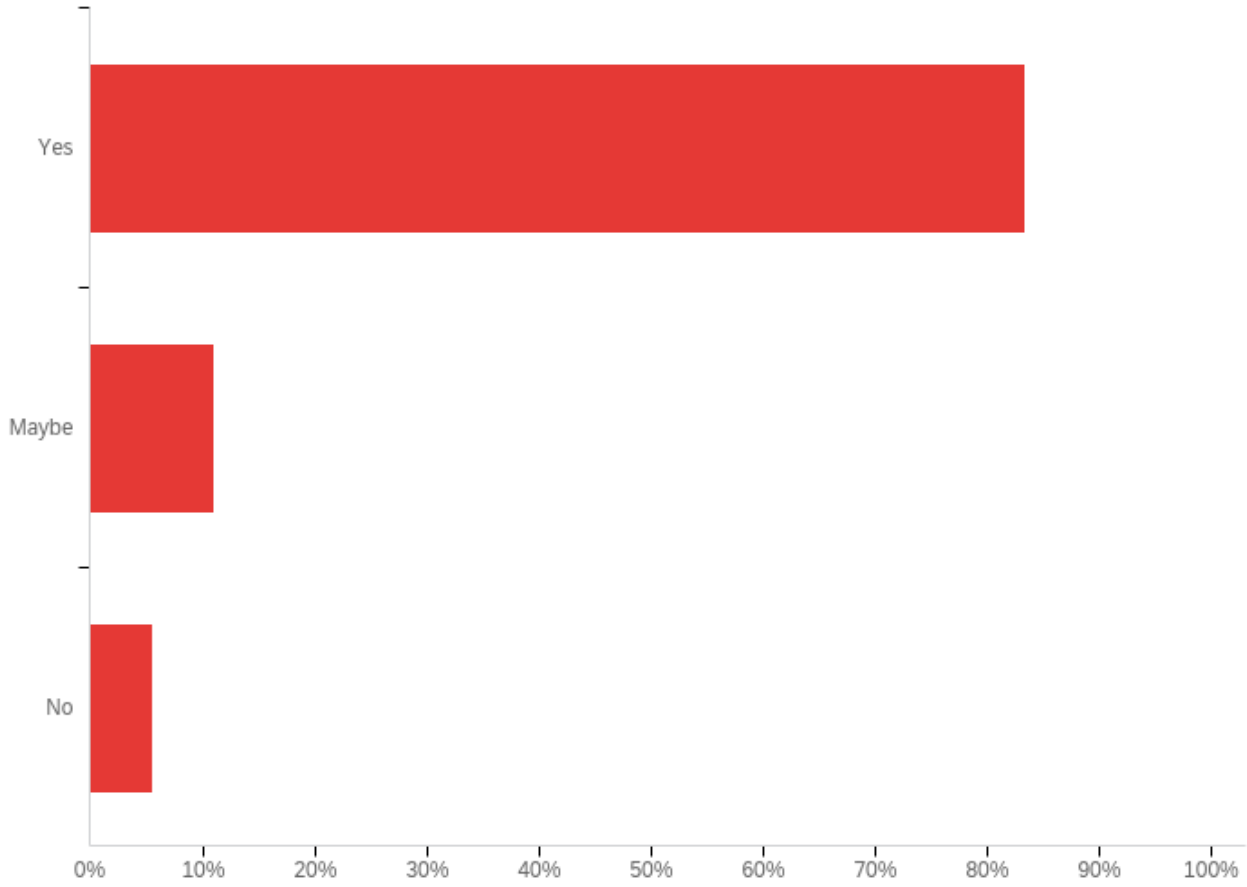
Q5: Enrolment was down 1/2

Enrolment in the course was down from last year, back to pandemic level. That seems strange, as last year we were back to pre-pandemic level. We'd be interested in knowing more about students' reasons for not taking the course. If you're aware of reasons why any fellow students who have the skills but shied away from the course, please let us know!
I know a few students who would've been able to complete the course but shied away from it due to the significant amount of work involved for standard credit
only 6 units of credit for a course worth 12
Don't know the exact reason but this semester, I know some people had trouble forming their groups. Since M0 is due on Monday and M1 is due on Friday the same week, it didn't give groups enough time to settle down (especially if people decided to drop out and people needed to re-group). I suspect some individuals who got left on their own may have dropped the course because there wasn't enough time to find a new partner.
This course is an enormous commitment and balancing it with more than a couple of the following is stressful: other subjects, a job, a relationship, family, health (including sleep), hobbies, a social life. While I look back positively on my experience and I would do the course again, I can understand why people avoid it and would actively discourage people from taking the course if they are not willing to make the necessary sacrifices to get through it. If this comment is made public, my advice to students thinking about taking this course is: treat the second half of the course as though it were a full-time job (i.e. 40 hours per week time commitment).

Q5: Enrolment was down 2/2

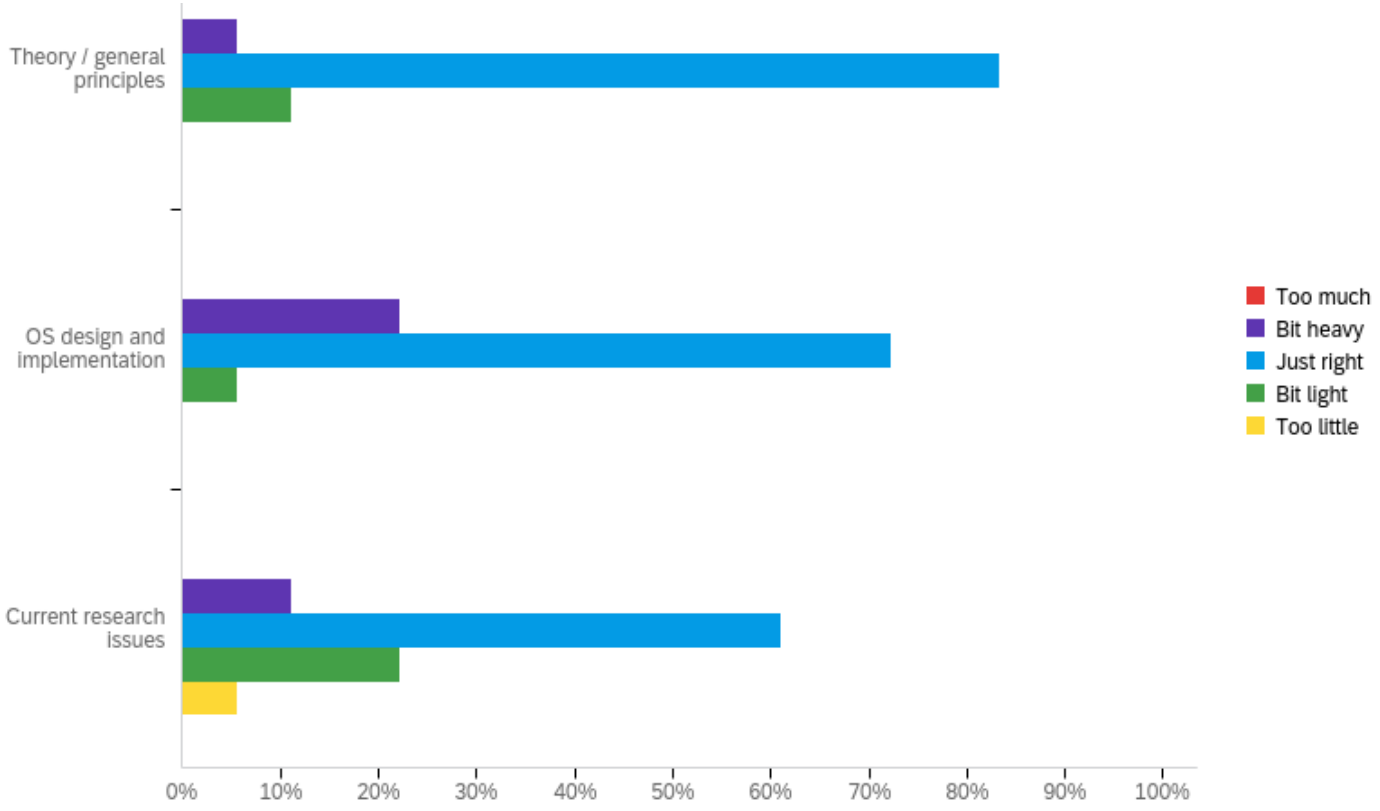
Enrolment in the course was down from last year, back to pandemic level. That seems strange, as last year we were back to pre-pandemic level. We'd be interested in knowing more about students' reasons for not taking the course. If you're aware of reasons why any fellow students who have the skills but shied away from the course, please let us know!
Less people are aware of the course. I think the 2 years of students starting uni in the pandemic just had less exposure to the world of CSE and consequently fewer people know about it. In my experience most people I have spoken to had no idea AOS was a thing (3rd year and below).
The reputation of the course as being extremely heavy on the time commitments just meant a few friends couldn't take it due to other commitments

Q6: Would you recommend this course?



Q7: Content Balance

The course is heavy on design and implementation issues. It also tries to remain close to present research issues. What do you think about the content allocation?



Q8: The best things about AOS? 1/3

What were the best things about this course?
The challenge of sinking your teeth into a large and challenging project throughout the term. Being able to learn from experienced teachers in their areas of study.
Gernot teaching us to analyse, understand and critique actual papers, very useful skill and engineering work.
The lectures are incredible. Also Mitch is an amazing demonstrator.
Bonding over shared pain with fellow students Ability to discover how to create a system by yourself
Lab sessions and talking to other groups about the project
For lectures: getting a good overview of systems research and what ideas have been tried etc. Also the content was almost all really interesting! For assignment: getting deep into actually implementing most of an operating system.
Building an OS for the project. Interactivity with the tutors. Understanding and experiencing first-hand how microkernels work and what makes them good/bad.
The community with fellow students

Q8: The best things about AOS 2/3

What were the best things about this course?
The highly practical nature of it. It is the most challenging (and rewarding) project I have completed in my degree.
The project, the lectures were also interesting.
interesting lectures the exam format was unique and generally enjoyable discovering that my body can function on a diet consisting entirely of coffee having a dedicated in person lab
Lectures and project. Guest lectures were especially fun and spiced up the course a lot. Community aspects actually were a thing, unlike all my other courses so far
I personally enjoyed the exam the most - reading and analysing papers really helped me to get an idea of what research is like and involves, as well as what constitutes "good" and "bad" research. It was surprisingly fun roasting papers and discussing them with others, I wish we had more exam prep/discussion sessions and was disappointed I didn't get to go through more past papers. Reading papers was also a large part of what made this feel like an "advanced" course as it felt like I was able to start thinking about how to push OS boundaries, and has been really helpful for me to decide whether I want to do research/honours and what in.
By the end of the course it is clear where to look to further your knowledge in OS.

Q8: The best things about AOS 3/3

What were the best things about this course?
I think I have become a better software engineer tackling the project, got better at debugging, learnt why best practices exist (sometimes the hard way). It's probably the course where I have learnt the most by far, about many things in operating system, but also more general software skills as well. All the lecturers and tutors are great.
The lectures and the main project
Challenging
The chance to get my hands extremely dirty building a complicated system

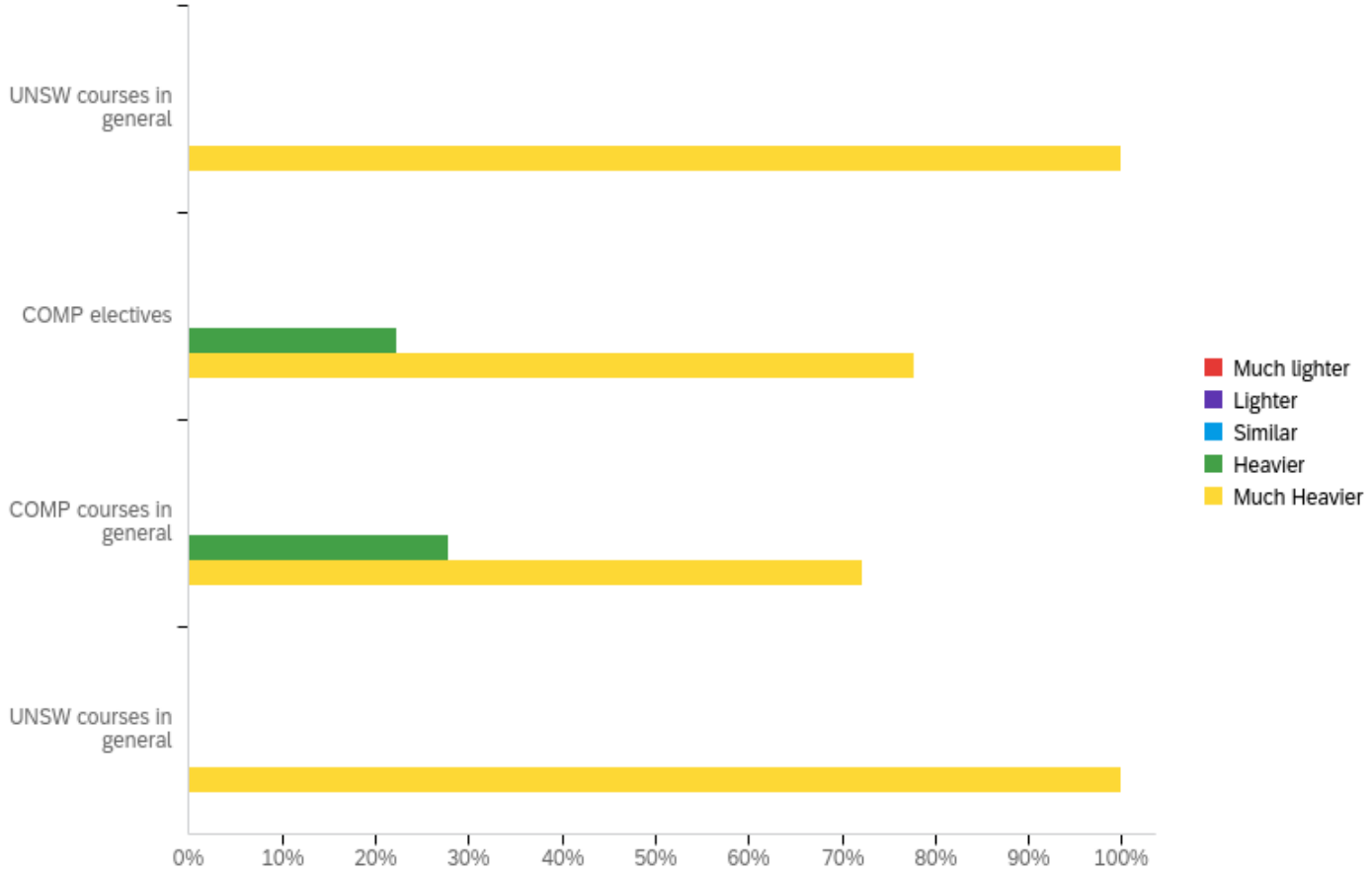
Q9: The worst things about AOS? 1/2

What were the worst things about this course?
The workload in the second half of the course ended up disrupting other courses.
Having to code through Vlab. It adds unnecessary difficulty to the course in my humble opinion.
Project suffered under trimesters - this should really be a 12UoC course. Can't recommend to most people with a similar workload to me solely because of the project.
Some disorganisation with consults/marking
The exam - it should only be 1 paper
Workload (as probably expected). Additionally, the lectures not really being related to the assignment that much.
Some of the milestone specifications could have been slightly more detailed. While most of our questions were eventually clarified by the tutors, a lot of the details could have been ironed out in the specifications. I found the FAQs section for the assignments in COMP3231 very helpful.
When the odroids would crash T T T T T T T T T
The highly practical nature of it. Debugging became a nightmare on several occasions and while the tutors were helpful and supportive, it is very difficult for anyone to help with bugs in our bespoke implementation.
Dealing with odroids over the internet. each milestone involved 2-3 occurances of my odroid unavoidably being dead for hours at a time.
Takes up a lot of time

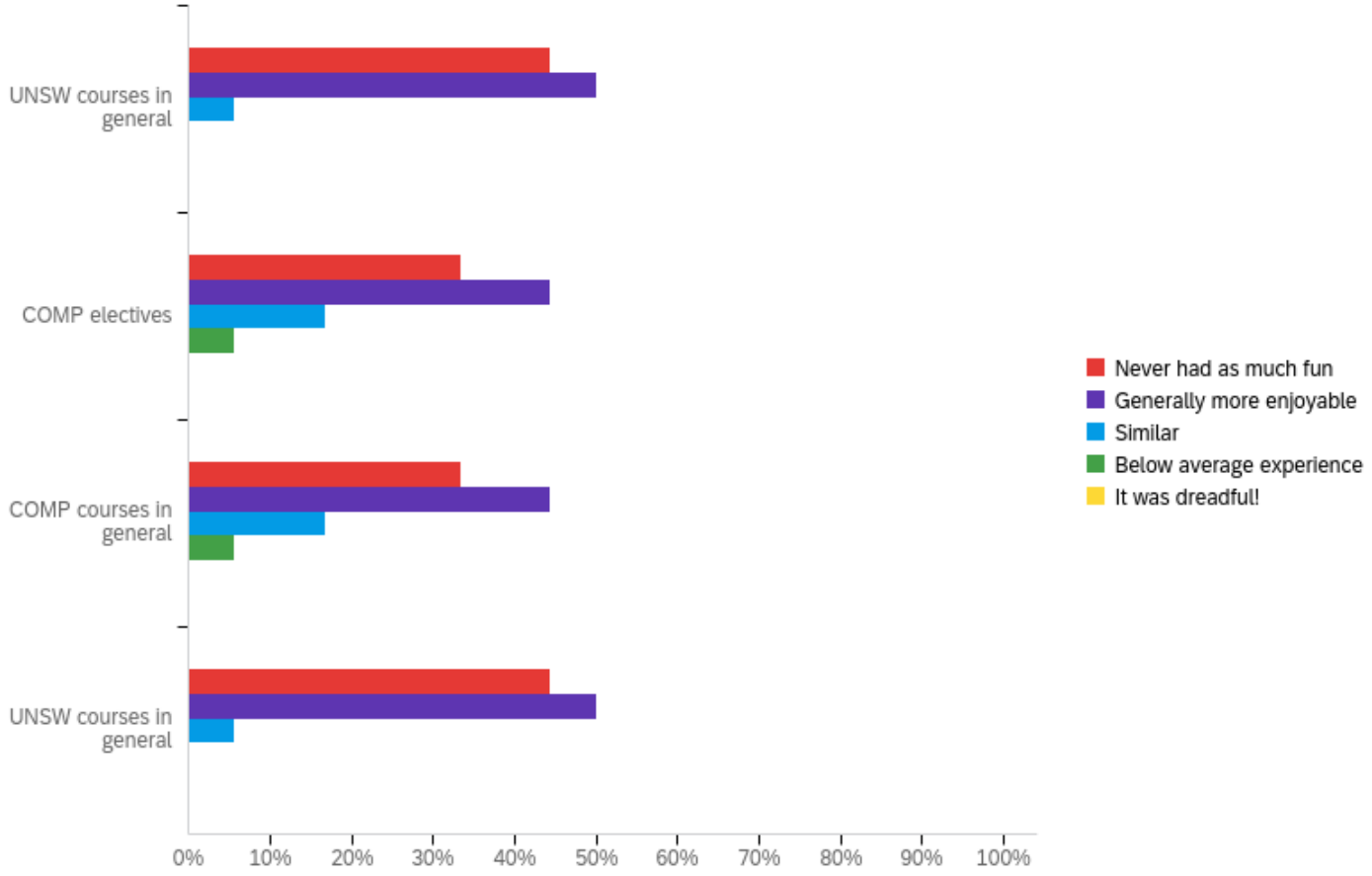
Q9: The worst things about AOS? 2/2

What were the worst things about this course?
Some of the timelines were pretty tight, especially at the start and end of the course. Additionally, I think that while the given code is supposed to be open for our own designs it seems to heavily favour event based systems. I think it would be good to provide synchronised versions of some given code as otherwise you habe to modify it pretty deeply or make some awkward workarounds
project took a ridiculous amount of time and overshadowed literally everything else i was doing didn't feel prepared for the exam at all until the short revision session a few days before
I think the documentation and mechanics of seL4 could be much better. I felt like I was always only able to connect the dots fully looking backwards, and lab sessions felt way too short - often the tutors would stay for up to an hour afterwards to keep talking about OS, but it was difficult to get tutor time during busy sessions. I feel like discussion is a really big part of why I enjoyed this course so I was disappointed there wasn't more time to do that with other students, tutors, lecturers, etc with so much to do and so little time.
Tooling sometimes break, seL4 reference manual has information missing (TODOs), really got tripped up by demand paging, lack of sleep (probably my own fault though).
Solving paging/multiprocessor bugs
The steep learning curve (but that is also one of the best things)
The lack of time to explore extensions or more advanced components (but that’s not on the course just the lack of time in the term)!

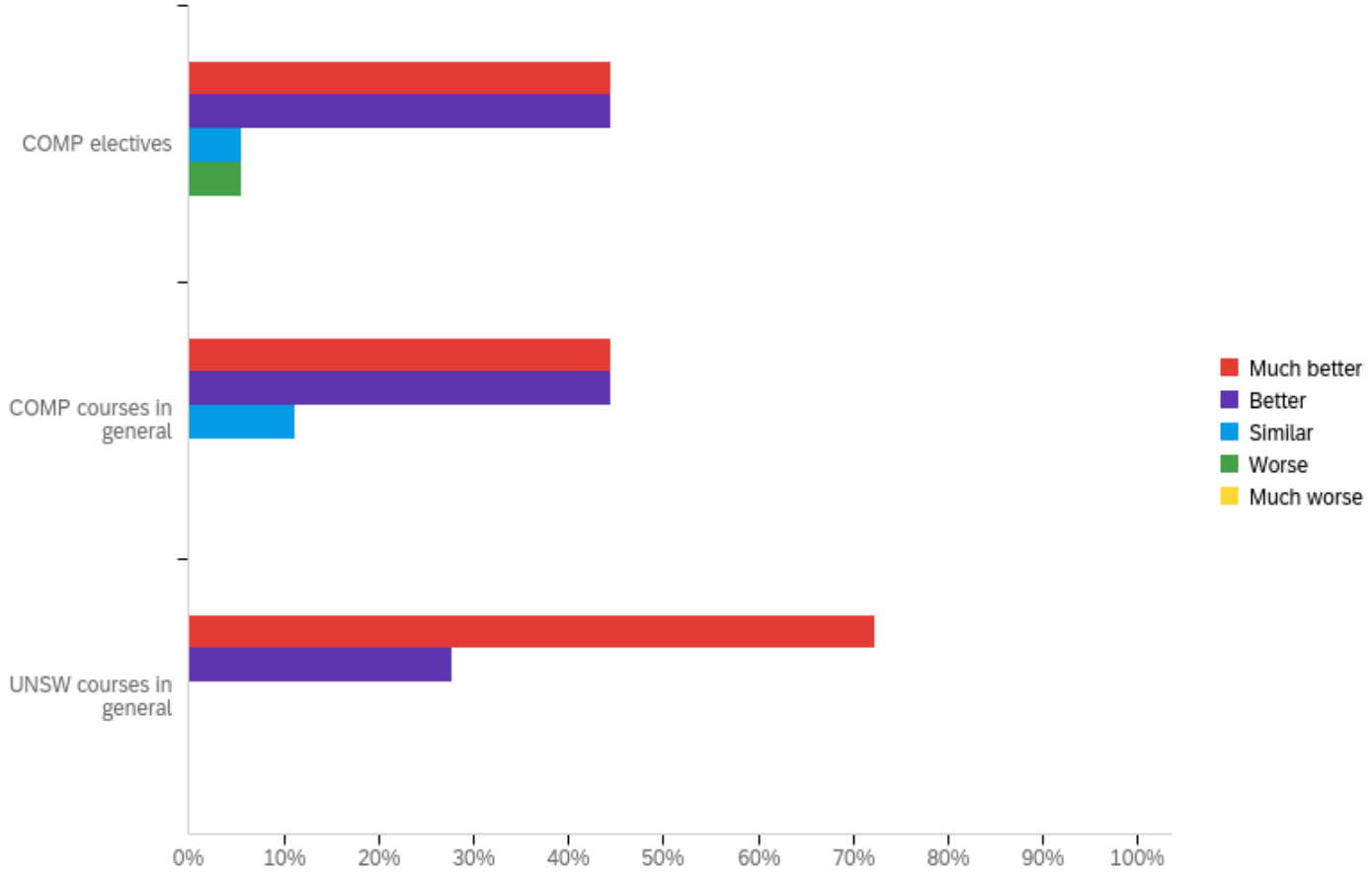
Q10: How does the workload compare?



Q11: How your experience compare?



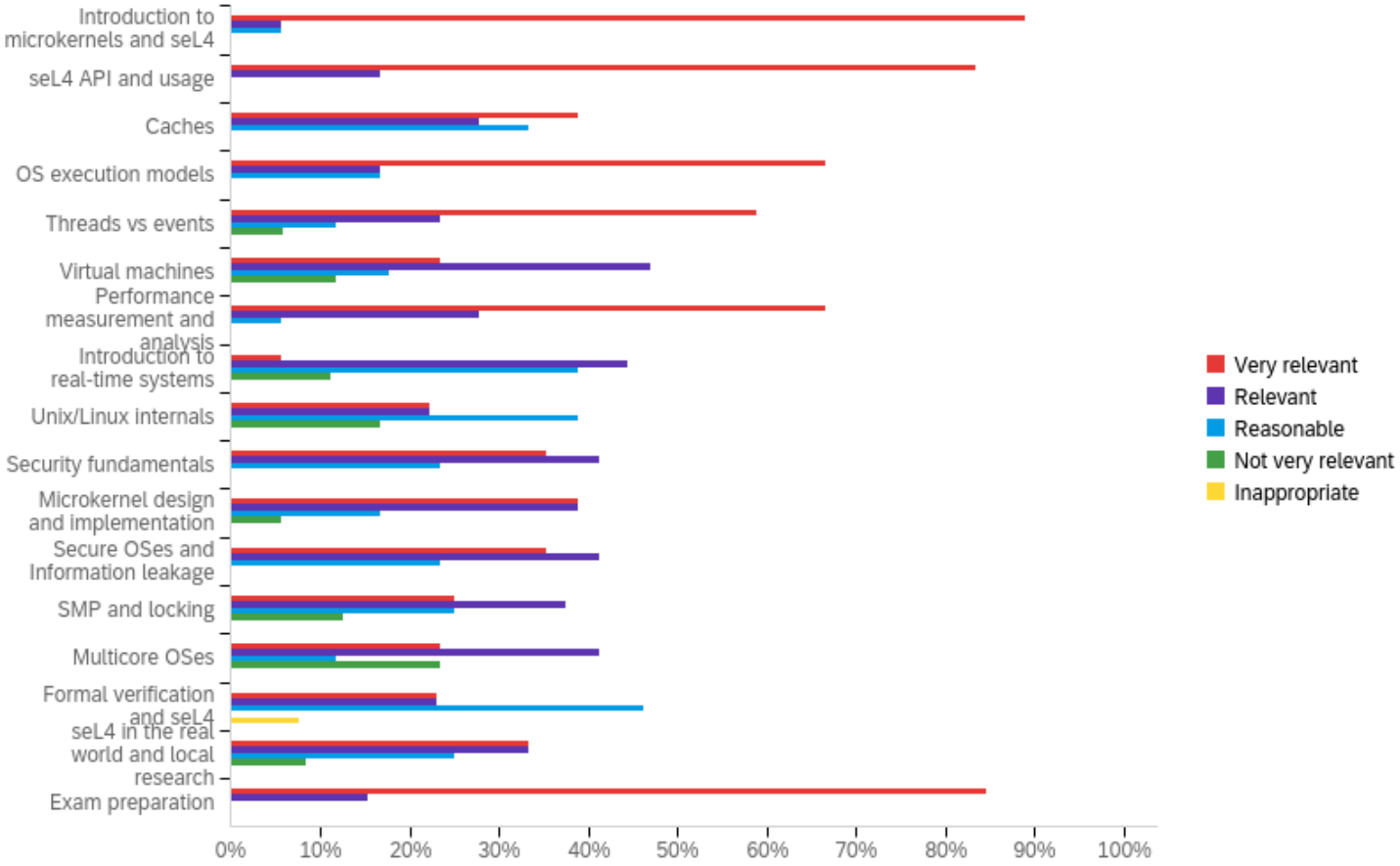
Q12: How does quality/value compare ?



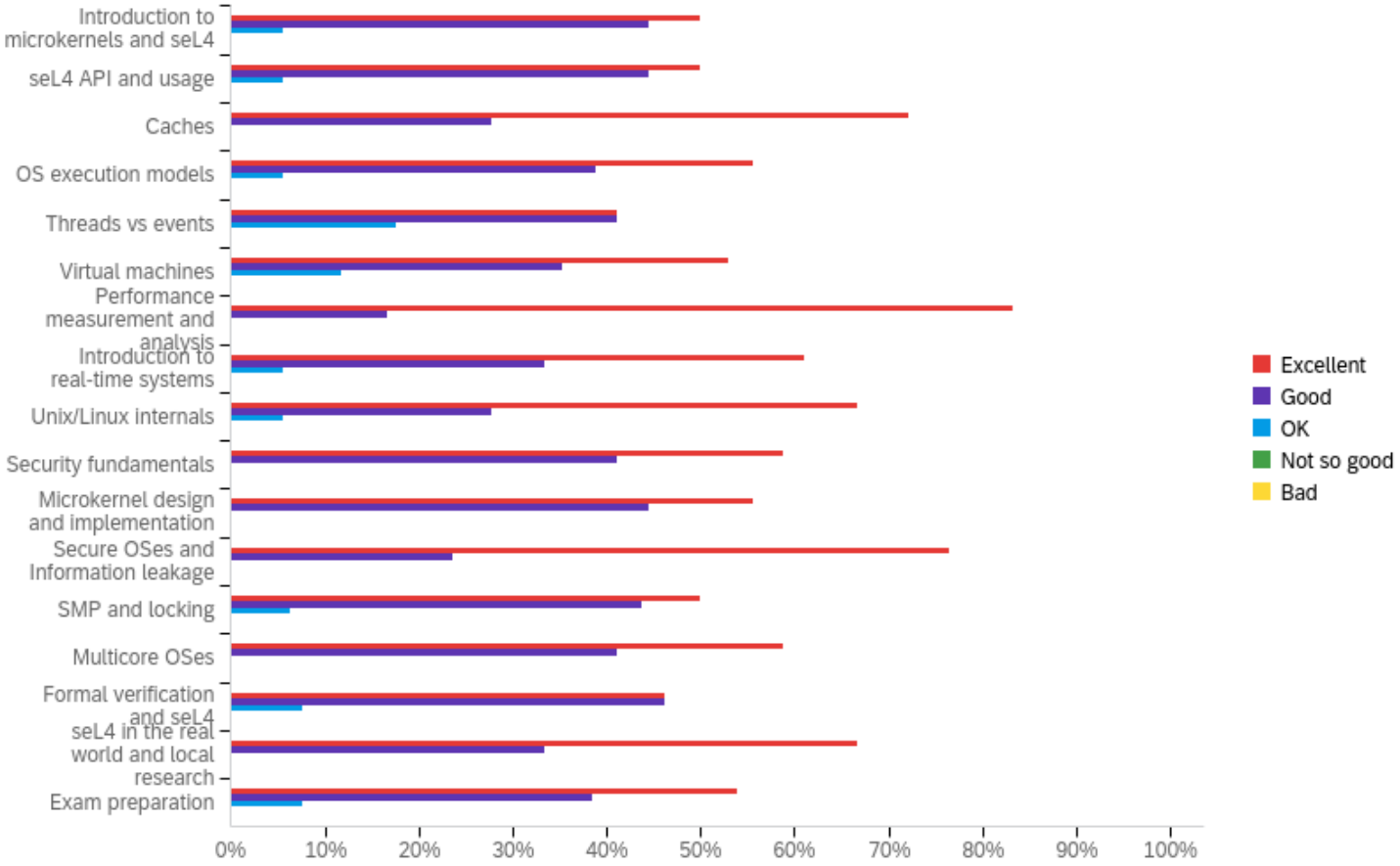
Q13: Required background

What background knowledge do you think you were missing that would have helped you in this course? Is a distinction grade in COMP3231/9201 a suitable preparation? Is it too harsh?
I think more c experience would have helped. Familiarity with standard OS implementations e.g. Linux would have helped since SeL4 shares many of the basics. Not too harsh at all. I got a HD in COMP3231 and still struggled in AOS =.=".
I felt like I was starting from a bit behind despite getting HD in OS as I hadn't done any work/research of OS in my free time. Some more SeL4 info would have been particularly useful.
The course requirement is reasonable
I personally had weak C skills coming into this course and that made the first few weeks more difficult than what they should have been. Though by the middle of the course my skills had improved. Perhaps some kind of C skills test that students could take in preparation for the course would help them get off to a solid start.
Yes, distinction is reasonable but it seems most people had a HD.
Distinction grade in OS is suitable
Current prereqs seem fine

Q14: Topics relevance/appropriateness



Q15: Quality of the lectures



Q16: Most useful material 1/2

Which material do you think will be most useful to you in the future?
A bit more in-depth for the first two lectures, as I was fairly lost during the first week of the project.
I'm working in performance so the stuff about measure performance, "benchmarking crimes" will be very useful.
Unix/Linux internals
Security
Things that directly expanded on the base course (i.e. caches, VMs) and general knowledge from applying ideas to the proect
The lectures on multicore and virtualisation felt really useful since they're more "mainstream" and things you'll likely encounter in just general software development work, as opposed to e.g. security/formal verification, since what's covered still feels more restricted to certain niche areas. Learning about benchmarking crimes was *really* useful. *nix internals and SMP/locking felt useful but only if you're working on the kernel or other systems with similar requirements.
All of the materials great. Microkernel design and implementation was especially insightful.
Things like caching and multiprocessor. Just because this sort of hardware awareness is not standard knowledge, and software/ISA stuff is easier to come by without prior knowledge of what to look for.
Probably a short introduction and overview of each milestone. This would enable discussion about some of the key issues as a class.

Q16: Most useful material 2/2

Which material do you think will be most useful to you in the future?
I think potentially more content directly applicable for the project but it's not a big problem
Caches, secure OSes and information leakage, multicore OSes
Caches/security/ other core OS theory which applies to everythng
I think virtual machines, benchmarking, and security would be what I see myself working with in the future I probably won't use these as much, but I also enjoyed learning about realtime systems, execution models, and formal verification
The list of references.
Performance measurement and analysis
Benchmarking, Virtual Machines and Caches
The cache lecture (since I'm going into low-latency areas). The Information leakage lecture was also incredibly interesting
All of it!

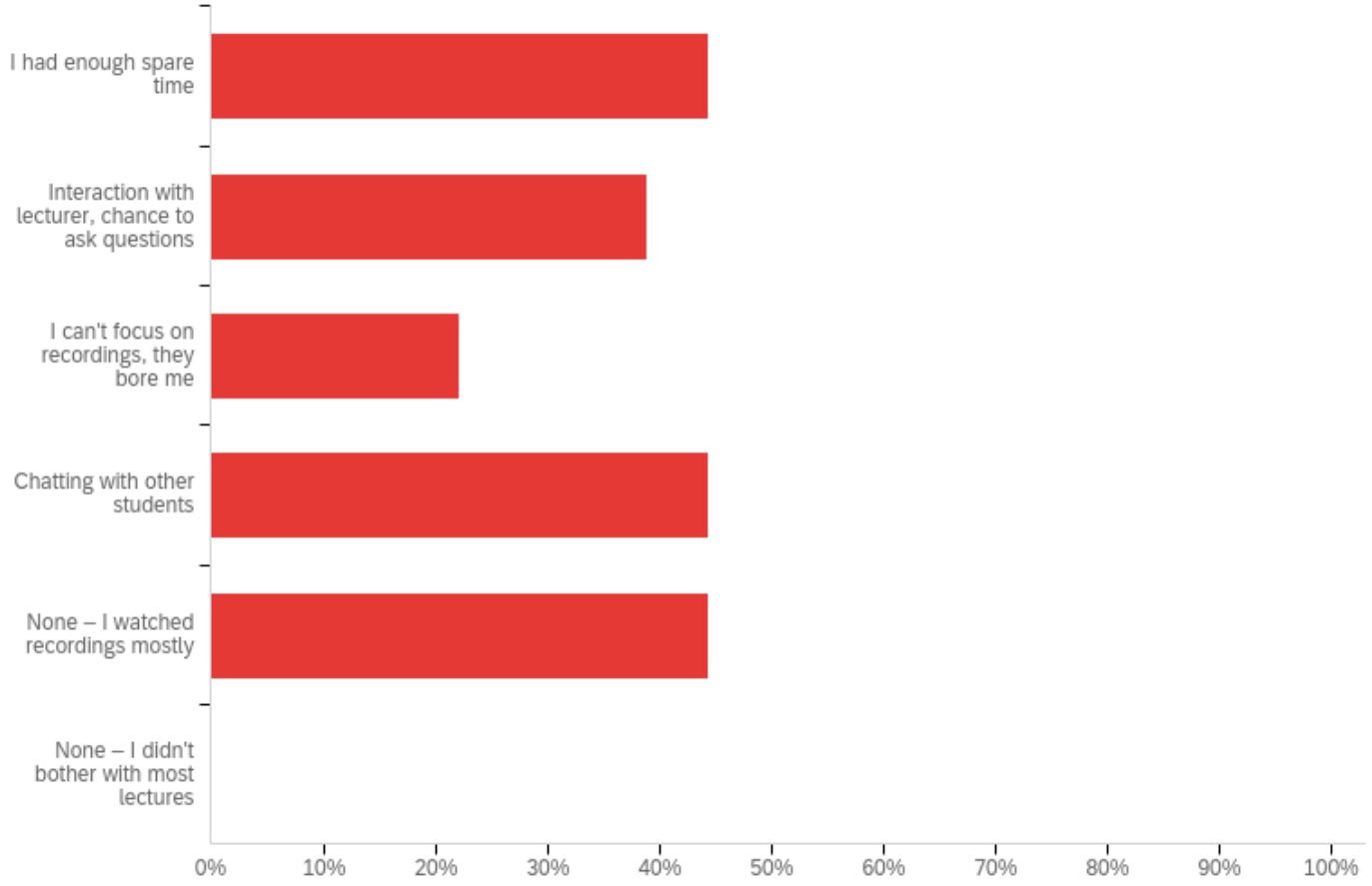
Q17: Missing material

Which material, not presently in the course, would you have liked to be covered? When commenting here, please also comment below, as we cannot add stuff without removing others.
More content on monolithic kernels (pls don't retroactively fail me)
writing drivers, device interfacing

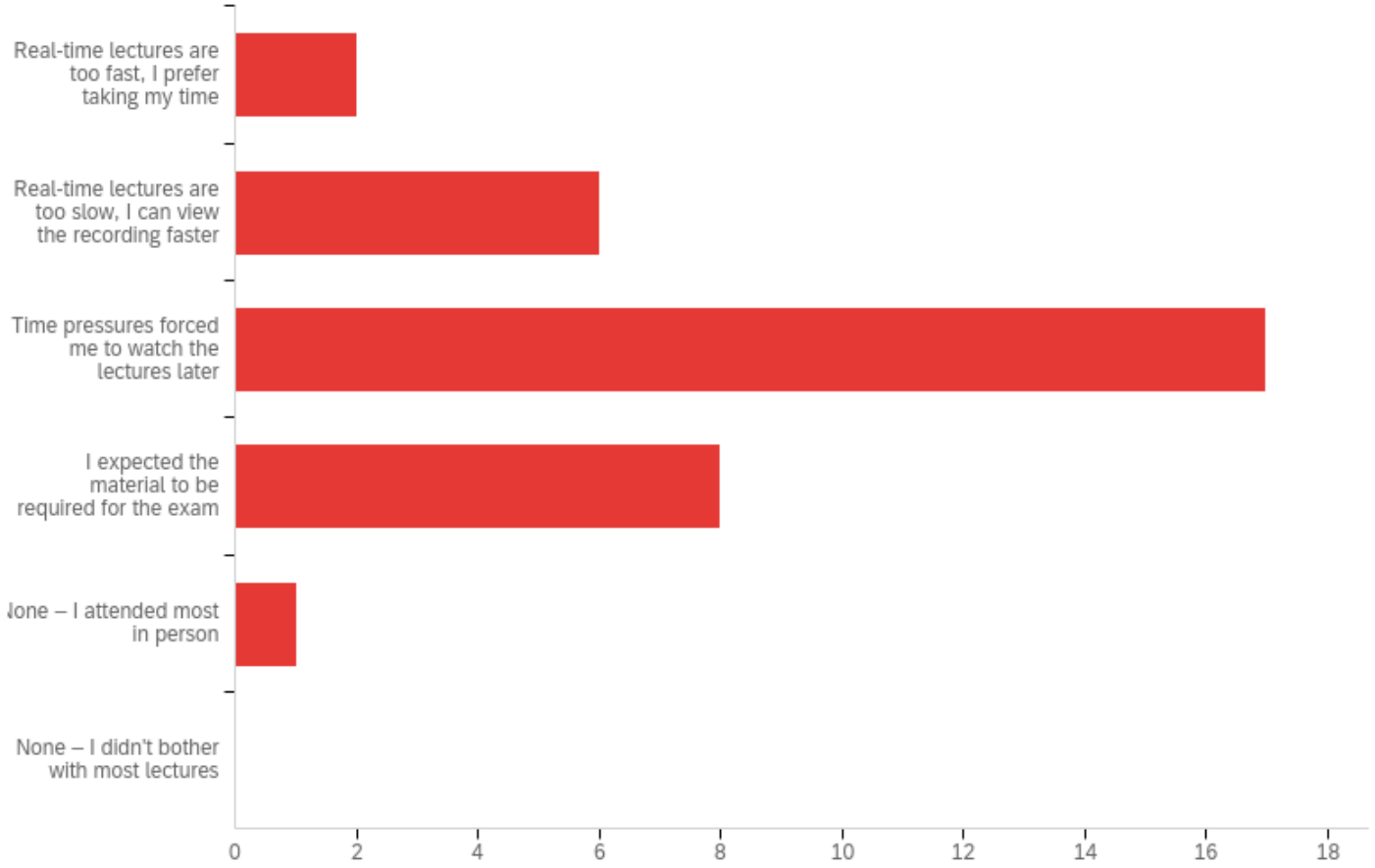
Q18: Material to scale back/exclude

Which of the current material would you like to see scaled back or excluded?
I understand why it's included but I found the hardware considerations (particularly locking) very dry.
Less lectures on things specific to seL4, that are hard to generalise to other systems/software?
Formal verification - while very interesting - I feel belongs more in the formal verification courses than in the OS design course.
SMP stuff wasn't too relevant

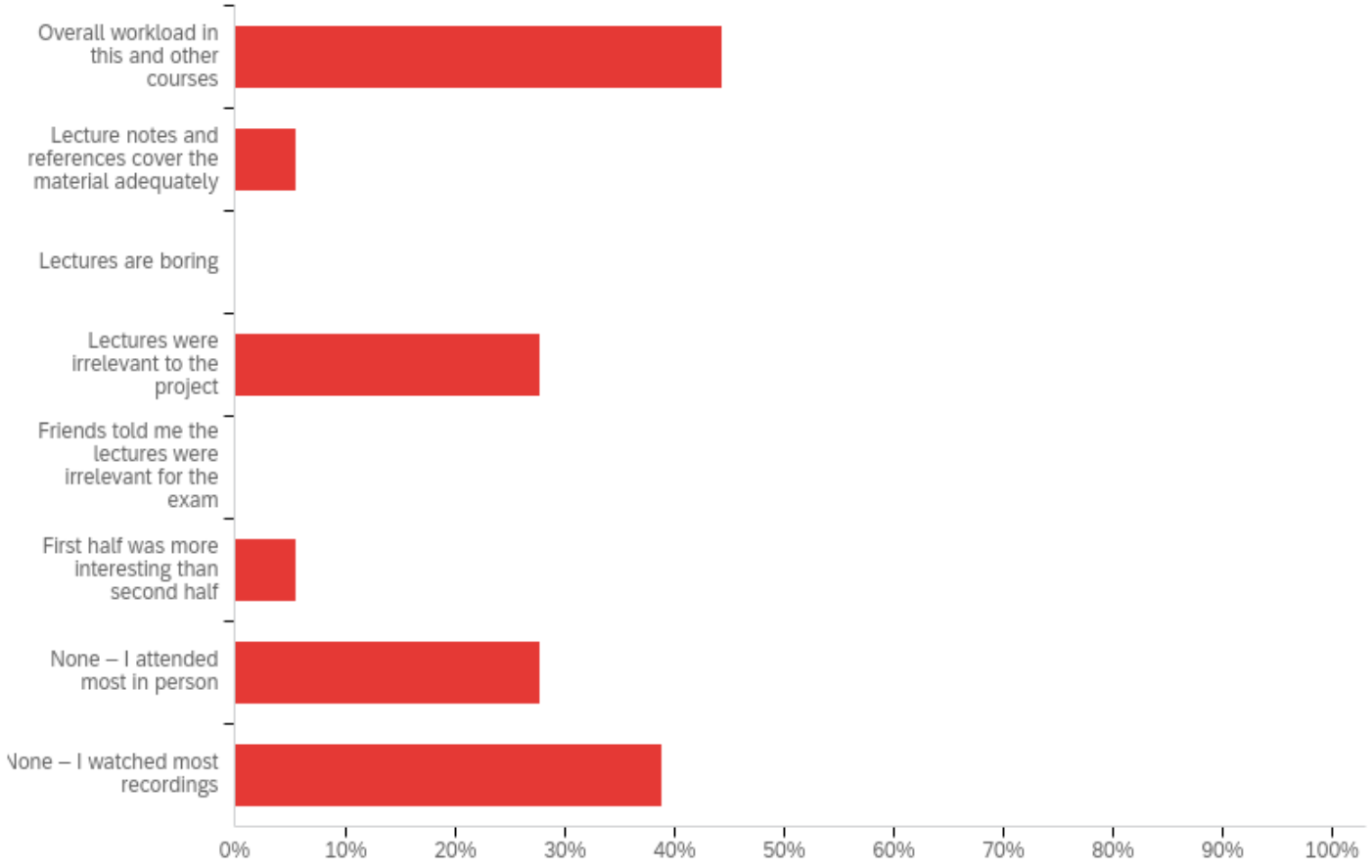
Q19: Why attend lectures in person?



Q20: Factors for watching recordings?



Q21: Reasons not to watching lectures



Q3: Encouraging in-person attendance

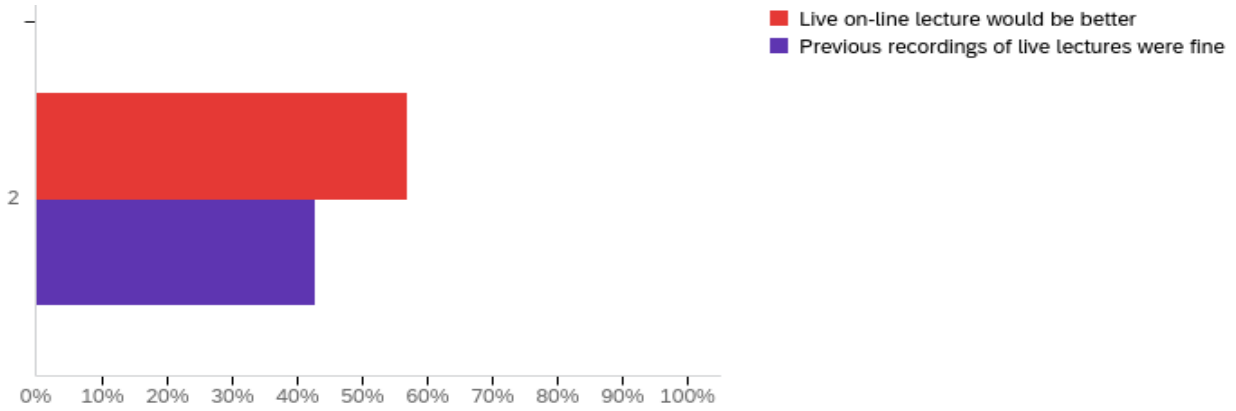
What would encourage you to attend in-person lectures?
Chance to ask Gernot Questions after the lecture
not having to stress about the project at the same time
If the lectures were more relevant to the project. I understand why they're not so they often took a backseat when under pressure.
the ability to pause in real life
Having them pretty close to milestone due dates meant that I often worked on the assignment instead of going to the thursday lecture. Also just general scheduling things: I had work during the tuesday timeslot so could never come, and a tute right before the thursday one.
I rarely prioritised attending lectures over working on the project. If the lectures were more directly related to the project, I would have prioritised them more evenly with the project.
If I didn't have work on those times. Also the Thursday slot often meant we were doing the project instead.
more relevance to project / not constantly feeling too crushed under the project to attend lectures
If there was a small section every lecture with advice regarding the milestones (non-recorded so people have to go in-person)

Q4: Other factors not mentioned above?

Other factors not mentioned above
Intensity of the project and deadlines made it difficult to attend all lectures.
EOS taught me some interesting things about synchronization and hardware-software interface
It would help if labs were scheduled before the lecture as at times, it was hard to avoid temptation to skip to work on milestone demos. The social aspect of having an informal discussion afterwards was a big factor for me always trying to attend thursday lectures.

Q22: On-line live lectures vs recording

While we planned to do all lectures in-person, the pandemic interfered again and people were unable to lecture and we put up previous years' recordings instead. Was that the right call or should we have done real-time on-line lectures? (Note that wasn't possible when Kevin had lost his speech and Gernot could not jump in.)



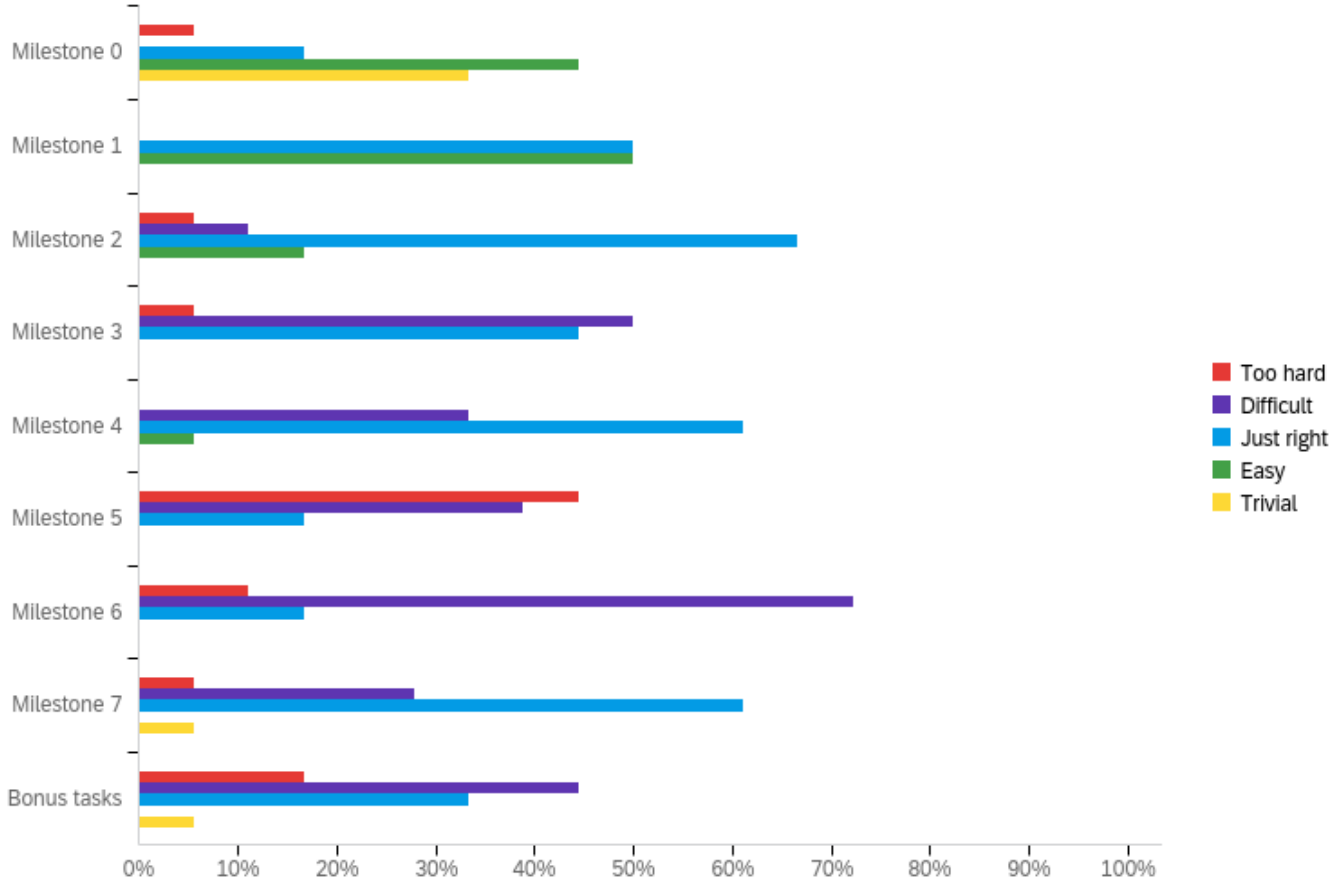
Q23: Other comments on virtual lectures?

Any other comments on the virtual lectures?
Audio levelling was really bad. Need to run a compressor + leveller in audacity on whole recordings to make them ok

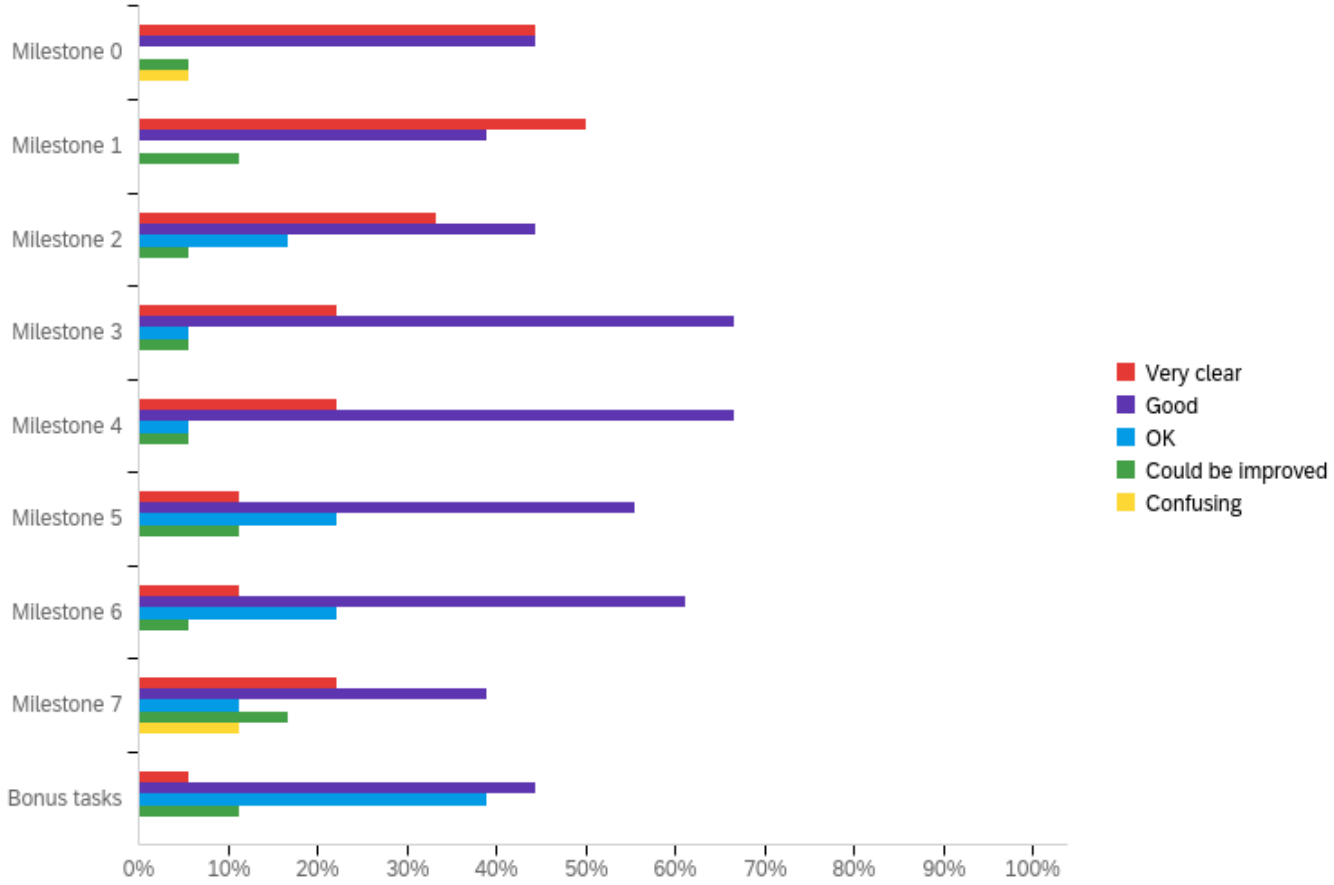
Q24: Other comments on lectures?

Any other comments on the lectures, especially suggestions for improving them?
More guest lectures would be really cool, even if that meant extra timeslots (i would happily come in extra times for them)

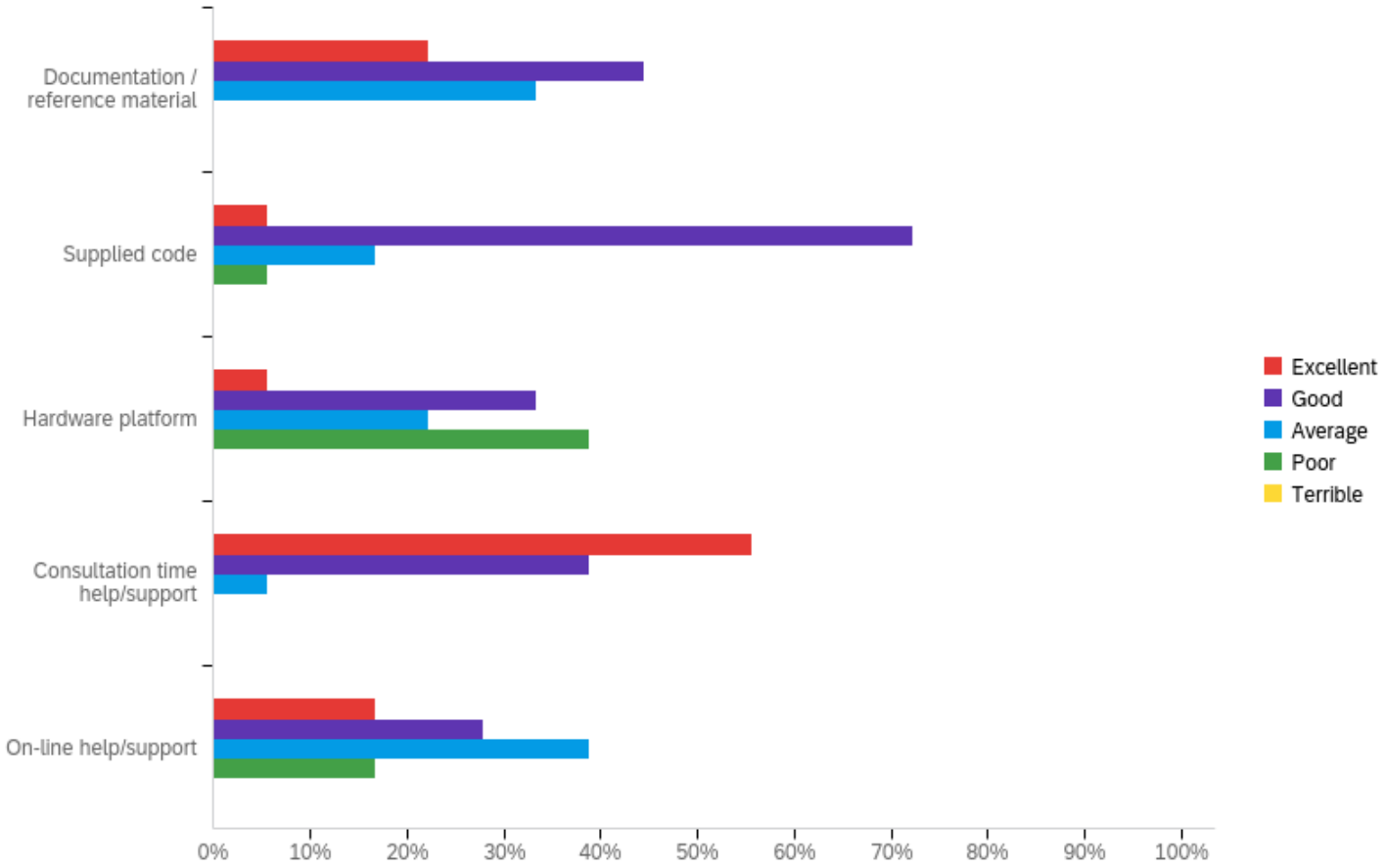
Q25: Difficulty of various project parts?



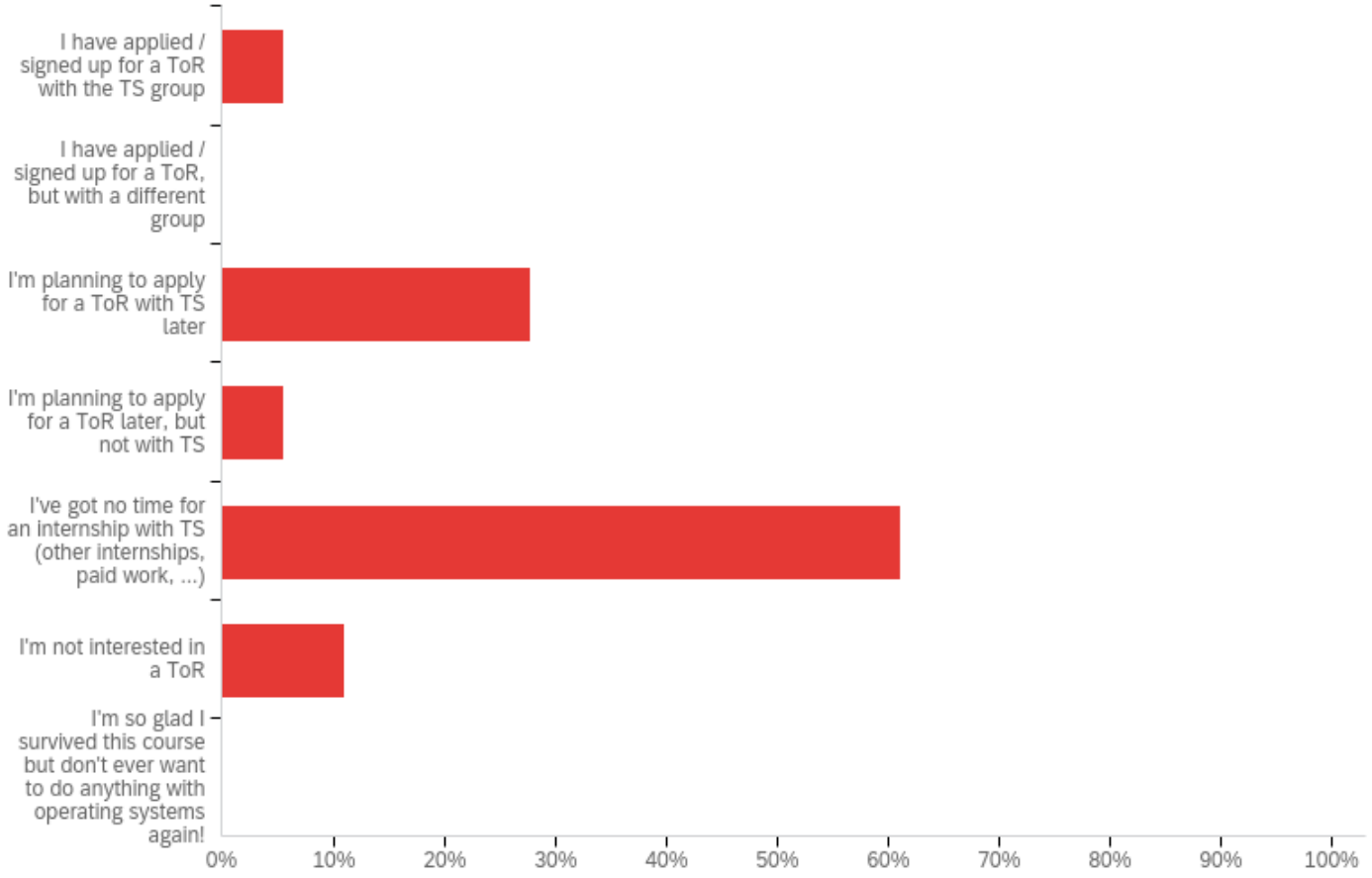
Q26? How well was the project specified?



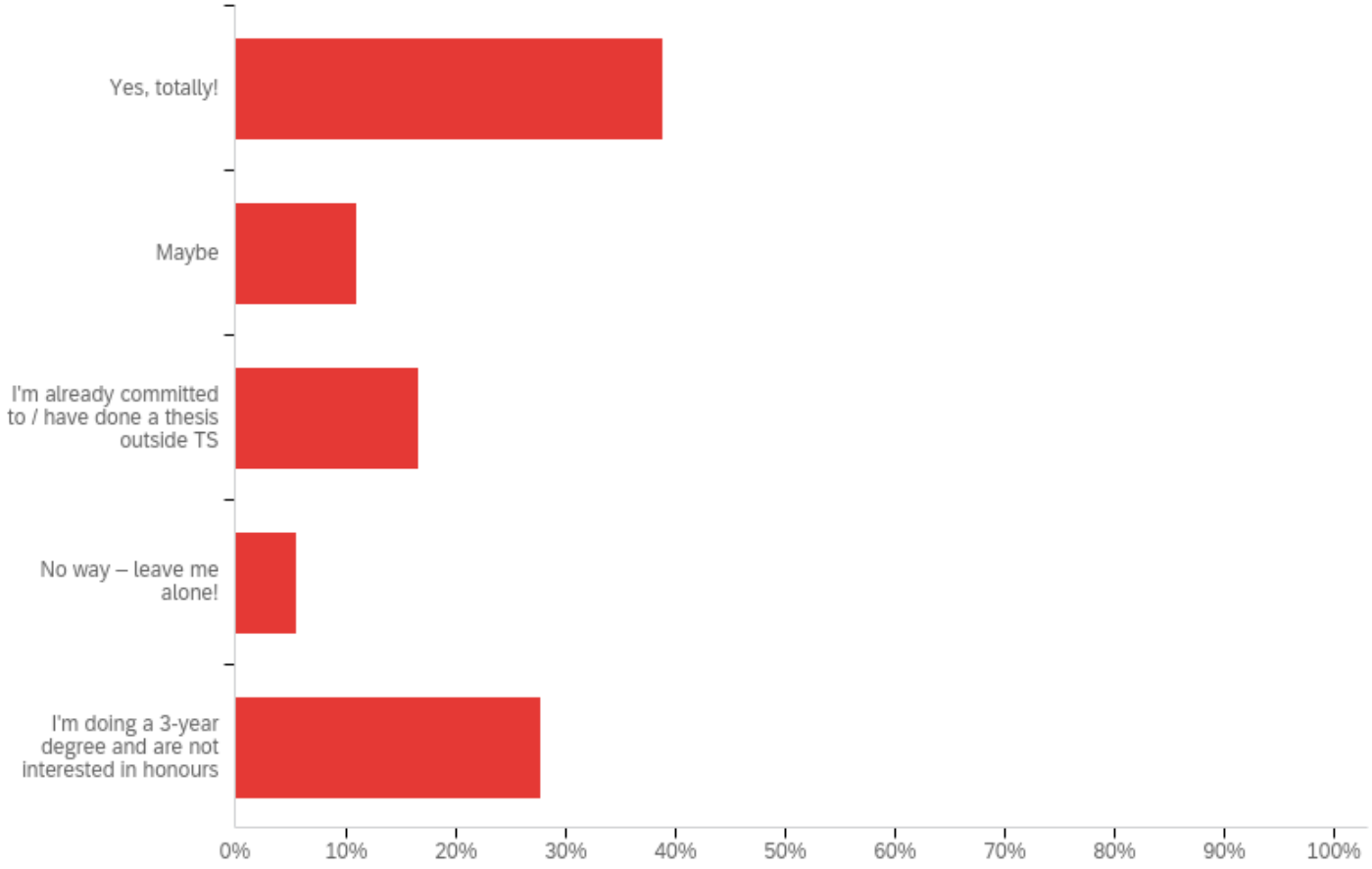
Q27: What was the quality of...



Q28: Interest in Taste of Research with TS



Q29: Interest in Thesis with TS



Q30: Comments on TS ToR or thesis?

Any comments on TS internships or theses?
I'm interested in doing a thesis as a Master's Student, from my understanding we're not eligible for ToR as it's for undergraduates.
I'm already doing honours in Chemistry and I really don't want to be at uni for 6 years plz
I'm not sure what the workload is (or how flexible it is), since I'm interested but already have a part time job & other commitments.

Q31: Any final comments?

Any final comments you would like to make?
Thanks Gernot, Kevin and all Tutors!
Mitch is great
Thanks for the experience
I really liked the course overall, but doing it made me an overloaded system - my total utilisation exceeded the schedulability bound ;)
Overall, I enjoyed the course! Thanks to everyone involved for a great term.
Thanks for such a cool course. Despite feeling like I was dying this term due to my workload and outside circumstances, I loved taking this course. 💜
shoutout to mitch, best tutor
AOS has been really rewarding and a highlight of my degree :)
this course was the most fun I never want to have again
Was an amazing course to take. Probably my favourite up until this point.
What a wonderful, once in a lifetime experience. The only sad part is I don't know if any other course in uni will ever live up to this course again. Loved every moment of it, wish there was another continuation of it (perhaps advanced aos :))) but I guess that's what TS is

The End