

- I think he is pretty good
- I enjoyed the lectures. They were very interesting, and I felt like I learned a great deal of conceptual OS knowledge from them (and from reviewing the notes online).
- Lectures were very fun and I learned alot
- ops-class.org website, test161 webui + client, previous year's youtube video's, numerous course staff assistance
- Assignments are difficult, but very educational.
- The content of the lectures are good and relevant. The assignments are way too hard for students and almost useless.
- I thought the lectures were interesting
- Both the lectures and the assignments taught me more about computer systems and programming than any other class I'd taken at UB.
- The work is very hard, but incredibly engaging. This course is definitely a real-world prep course and goes beyond the classroom, teaching time management, problem solving and different types of thinking. The lectures were very interesting and the course website is fantastic!
- The videotaped lectures, notes, and slides were awesome. The lectures were pretty entertaining and fun to watch. They definitely help me study better and I appreciate having all the extra material to use. When you put coding snippets into the lectures, that was great to help me understand synchronization concepts
- TAs and ninjas are very helpful. Many work overtime for this class... Lecture slides are well organized.
- Having a lot of office hours is nice.
- Lectures are great.
- The pace of the material was very reasonable. It was not too quick that I was confused, and it not too slow to be boring.
- This class was awesome, one of the best at UB. The course was structured well, covered the expected topics and taught me a lot. The projects were challenging, but necessary for understanding of the topics covered in this course.
- Awesome assignments. Made the semester worthwhile
- The lectures were fun and really never felt dull.
- I found the tool chain test161 to be quite effective. It was easy to use (for the most part) and was a way to get feedback and gauge progress through the programming assignments.
- The lectures in 421 were incredibly stimulating and deep. Every topic was thought provoking, and the level of difficulty throughout ranged enough such that each student in the course was allowed both moments of clarity and analysis, and moments of challenge and intrigue. The course website was an extremely useful tool, as it served as an easy-to-use hub for all things CSE421. Accessing schedules, lecture notes and videos, and the course forum was all doable from navigating via ops-class.org. The course staff was fully equipped to answer questions and supplement lecture material.
- Loved the progression from locks to system calls and then finally memory management.
- The project style of homework was much nicer than the weekly homework that we see in other classes.
- assignments are challenging
- None.
- Lectures were always interesting and led into topics covered in the lab assignments.
- Having TAs, the website, classes recorded and put on YouTube, I really wouldn't change anything.
- Challen is a decent to middling lecturer. It's crap material and subject matter and to his defense, Kosar was actually worse at it.
- The assignments are definitely challenging enough and the lectures were always interesting.
- This was one of the best classes I have ever taken. I learned more about these various subsystems of operating systems and gave myself the confidence that I could contribute to developing a real production system.
- I found that the level of difficulty increased well. The first assignment was just the right difficulty to learn the system while still doing something useful. The consecutive assignments were of an appropriately increasing difficulty.
- Nice assignments..... Very less time.
- The projects were really good. I learnt a lot from them. TA s were really helpful.
- Assignments were very good! and extensive TA hours.
- Classroom Teaching and Recitations
- The assignments.
- The course videos were amazing, I really enjoyed attending every single class. The assignments were challenging and interesting. The evaluation methodologies were fair.
- Assignments were fun and learnt a lot from them

- The lectures were engaging and I feel as though the topics presented were interesting to learn about and applicable to my everyday computing.
- Lectures was fun to attend, and I particularly liked the fact that the course never went too deep into hardware aspects of the OS.
- The programming assignments were of increasing difficulty and are gave the right platform to get hands on experience for starting with kernel development.
- The assignments!
- A lot of TAs and Ninjas, going over topics in class before the assignments, previous exams. Tests for our assignments and a way to submit multiple times and make sure we get a good grade. A professor that actually cares to be teaching! Also thank you Ali and Xu for taking the time to grade our exams!
- Covered as many concepts possible.
- The class notes and the video lectures were very helpful.
- Assignments
- The projects are really intensive and have improved my debugging skills.
- A life changing course! I am a different person after taking this. And I say this because this course has made me more confident at programming!
- The professor's enthusiasm was excellent and helped me in relating with the topics covered in class
- Lectures are pretty entertaining + informative
- The ops-class.org site and YouTube videos were helpful.
- Hacking and fixing OS161 is probably the best(and worst) assignment I've ever done. I've really learned a lot from the assignments, learned a lot about operating systems from the fun classes (which are meticulously recorded) and GWA's musical tastes.
- The classes were really good, I could even watch the recorded version of class if I missed out on attendance. The assignments are tough and take a lot of time but totally worth it.
- Teaching style of GWA, insights into how to design system
- TA section is great than ever. It really helps.
- Git, Test161, Virtual Memory
- The lecture was intriguing and provided new information. The work was related to the course and provided good insight into operating systems.
- Lecture was stimulating
- Geoff is easily one of the best professors I've ever had. You can tell he is passionate about what he teaches.
- Lectures were very interesting. Recitation was helpful for completing assignments. Having recorded lectures and recitations was effective for studying, completing assignments, and reviewing the material in general.
- TAs were very helpful.
- The efforts put forth by professor and TAs is really good. They have designed the course structure and timeline very well. Also, one thing which I felt in love with was TEST161. It is the test suite I have never seen before. They have worked very hard for that.
- Online outlines, slides, and lectures were very thorough. It it incredibly useful for studying or going back to master material. I think every course should be required to provide similar resources. The ops website is really great.
- The programming assignments had insane test cases and was so much FUN! Loved the course website and all the work put into it. So much effort has gone into making this class awesome. Really appreciate it! Loved that we didn't have to submit our design documents and all points were for programming only. Adding deadlines to projects helped too. The BEST course at UB!! Lot's of help available if you need it. Don't get intimidated about taking the course, dive right into it.
- I found the recitation slides and TA help to be the best help.
- The lectures were very interactive and engaging.
- The lectures were quite informative and easy to understand, and being able to actually write an operating system was quite the experience
- I liked the lectures/videos/notes, the course website and assignment submission/feedback system is stellar, and Geoff is very enthusiastic about the course.
- really, really , really which I understood the os161 OS, still asking myself what program i am writing.
- Using OS161 was the best part
- The OS161 project and the amount of office hours provided for it was incredibly challenging but manageable. The heavy weight on exams allowed good leniency so the project wasn't one make or break grade.
- The assigts were great
- Everything... The professor goes out of his way let everyone learn what he wants to teach.
- Recorded lectures.
- Geoffrey Challen's existence. Also, the gro0vy music that made learning invasive material late in the day less painful.
- PERFECT CLASS. COULD NOT BE BETTER.
- The concepts in class were taught pretty well.

- - The course material was challenging but interesting and stimulating enough to merit the work required. - Scott's test161 was a great tool with which to work. - GWA's lectures were enjoyable and informative. - The general attitude along the lines of "yes, this course is difficult; get to work" was refreshing. - Working with Discourse was pleasant.
-

- Design goals for ASST2 were underspecified. In particular, I ran into trouble with available memory varying between tests. Because of these moving goalposts, I ended up writing code to satisfy the tests without a larger picture in mind. Having a hypothetical real world use case for the operating system and more consistent system configuration for tests would have made the design phase more enjoyable. Even if the stated design goal was to write an OS capable of running on a variety of hardware, design decisions would have been more interesting and could have been made more confidently.
- Lecture time was wasted waiting for answers. A significant portion of lectures was spent waiting for students to volunteer answers to questions. GWA mentioned cold calling at the beginning of the semester, but it did not happen. Cold calling, with the proviso that "I don't know" is an acceptable response, would have minimized time wasted waiting for responses.
- A little more guiding would definitely be helpful, coding the portions of the operating system is quite painful when you have to find out where to start, which takes up a good portion of the time. I personally would rather have that time saved for debugging. Also, I personally think grading is slightly more unfair than fair because of how the automatic tests were made to work. Some of the tests require you to pass the other tests to run, meaning even if I could get the last portions of the tests to run, the actual grading system wouldn't execute it unless the first tests are completed. So in that sense, I wouldn't get all the points I fully deserved.
- Could be more like harvard... But there should be more accountability for partners that dont do shit.
- Obviously a challenging suggestion, but allotting students working alone additional time for assignments, as opposed to omitting particular components. The entirety of OS161 can be implemented ambiguously, and the projects are cumulative. In the later assignments, running TEST161 exposes errors in previous code, and debugging/redesigning your own work is much more challenging when only one person is involved.
- hm...so hard....
- This course did not feel like an intro course. This class has no prereqs for the material that was covered and the time commitment needed to do well was far outside of the range of being a 4 credit course.
- Anything else would be an improvement
- May be give more time for assignments
- More help in understanding assignments and how to start off with them
- More consistency in the OS161 project from semester to semester: it seemed that even some TAs were unfamiliar for new, more difficult requirements added into the project that went above and beyond what they had to do, making it harder for them to help.
- Maybe a bit more guidance where to find more help.
- More guidance/resources for the programming assignments would be helpful. The assignment descriptions and recitations do little to help understand the assignments. There are office hours, but I'd rather just have better documentation/guidance.
- The resources for the operating systems project were somewhat lacking and led to more confusion that there needed to be
- instead of spending the last few weeks doing nothing, more time could have be spend going over assignment 2 , which most of the student find difficult. doing this assignment brings in its own complication, instead of fixing your own coding error, you have to fix git problems, Mips error install errors. it would be wise to make a virtual appliance a self contain environment --not vagrant . Linux has soft and hard links put all the files that have to be modify for Assignment in one folder. You can't do this course without going for help from the Staff. which is not good
- The programming Assignments were complicated. There should be more focused documentation about tackling them
- One big problem I found is that even though the TA's were helpful, they unfortunately were one of the only ways for anyone in the class to understand the assignments and information given to us. I believe that on average, if you wanted to pass this class, you would have to visit the TA's regularly and I don't believe that is a fair way to get people to learn material. It is effective though, to just keep throwing TA's at the students for whatever questions the student has, but it seems like a lazy way to solve the problem of helping these students understand the material. I believe that code reading questions shouldn't count for a majority of the grade, but should be graded and improved on, as to give the students an incentive to do the code reading. Also giving good instructions that line out where to start on assignments are good to have. Idk what could solve this, this could also just be me complaining about taking this class along with 4 other CSE classes. I think im going crazy XD
- Although there were many TAs and ninjas I found that getting effective assistance was not something you could expect. TAs seemed to rely on their solution code, and just try and point out a mismatch with yours. This is hardly ever what the issue was. It would be more helpful if they discussed your design and strategy with you in order to understand your problem. Many TAs did do this, just not all of them.
- For someone like me, who has no background in OS or OS related concepts, the course is really tough. Professor might include some basic stuff in introduction to assignment material. Generally people followed Jinghao's blog for whole course so something informative can directly be added to assignment description. But again, these all comments are subjective. My comments are due to the difficulties I faced in completing assignments but others who were able to finish easily might not feel the same.
- This doesn't seem like an intro class. And the workload is a lot given we are taking a lot of other classes as well. I would suggest breaking it up into two classes. One intro with ASST1 and 2. And the advanced one with ASST3 and ASST4.
- Perhaps split the assignments into smaller "sub" assignments to facilitate completion because the workload for this course is large.

- The only thing I could think of is links to resources to help with the assignments.
- Projects seemed unrelated at times to the lecture. Tests were extremely difficult. Lack of organization and preparedness for the projects
- More information on the projects. The lecture would skim or not cover a project topic at all and there was very little help online or in books to help understand the material.
- one of the best courses
- I can't come up with any ! I loved this course
- The only suggestion I have is the work load. The assignments take way longer, spanning weeks, if you are taking any other challenging course(as in spring semester), it's hard to complete either and you are forced to choose one to complete.
- Please avoid last minute test script changes (forktest in ASST2)
- The assignments are TOO difficult. One cannot expect to complete the work while taking higher level required courses simultaneously. I've felt lost this entire time and I'm extremely disappointed in the cse program for making students feel so depressed.
- Slightly reduce difficulties on assignment perhaps
- Need some more help with reading material for the assignments
- I would suggest keep a test to determine partners at the beginning of the semester. I was stuck with a partner who barely worked after the spring break. As a result I suffered a lot. Still suffering while I'm writing this since I have had to do almost everything in the assignments.
- None. Extremely satisfied with the course. The best course in UB I've taken so far
- Lighten the course just a little bit. By this what I mean is the test cases could be made little less strict. Because with four courses it becomes very difficult to get full points.
- More guidelines on doing assignments could have helped. Instead of assignments only, there could have been quiz or bunch of smaller programming questions to solve.
- Why not make more information available for completing the assignments? Otherwise most people are asking the TAs the same questions. An updated blog would really help.
- As the course is evaluated primarily on the project work, more of the lecture time should be devoted to that material in a more explicit way.
- Assignment 3.1 was easy, and 2 weeks for it was too much. 1 week is more than enough for it. Some way to catch memory leaks in assignment 2 for process & file system calls would be very helpful, although I doubt it can be done. More penalty for late submissions. Only 10 points deduction for Assignment 3.2 is not fair.
- The work load is way too much for a 3 credit course. There were too many moving parts this semester to adapt to during the assignments. Often I found myself and other students passing tests and moving on to other work, and found on updating that they failed tests that they thought they were done with. Constantly pulling and updating was time consuming and created many issues with assignments.
- I would suggest more of guidance with the reading material regarding assignments.
- The course slides should be made available in PDF or other formats so that they can be viewed offline too.
- None, its a very good course.
- More help with Projects
- Nothing comes to mind.
- Give us more time for assignments..... without plenty.
- I think the system of assignment deadlines was not the most helpful. I think it would be better if the deadlines were changed to a sort of checkpoint. Each checkpoint would have a recommended completion date but students would not be penalized for not having the project completed at that point. Also, each checkpoint would consist of a set of points for different parts of the project.
- Probably tell people not to read pearls in life
- The only criticism that I have for the course was that things on the testing suite were very subject to change even the night before an assignment was due. This can lead students to believing that they have the correct solution even if they do not with the updated tests. Although this is how industry works and we were the first class to use the new test161 tools, I think that it would have been better if the tests and the submission tool were completed before the course had started.
- 1. Despite your thoughts on the matter, the work load for this course, in the form of the assignments is for the most part unreasonable. You do a good job of masking it behind your veil of: "we offer an army of people for you to go to for help". Any course which mandates office hour participation in order to succeed is fundamentally flawed. Yours is not the only class anyone is taking during the semester, and not all of your students fall into some archaic stereotype of having zero outside responsibilities. Some of us have full time jobs, families on top of our full time jobs, and let me reiterate: OTHER courses which also require more than a few hours per week of work. The only people I have witnessed being successful are those with the free time and lack of outside responsibilities allowing them to sit and pick at the cast aside crumbs (see 2.) from your staff. 2. I made it a point initially to make the couple office hours I could attend per week. I stopped shortly before ASST1 was due. TA's and Ninjas who cannot answer questions are not worth the time. I had one TA directly tell me that he could not answer my question because he had been told that he needed to be purposely vague in his responses. If I am coming to anyone for assistance in understanding a concept, for a course I am paying for, which the department mandates I take, I do not want to waste an hour of my life and time sifting through garbled vague messages as if I inquiring as to the meaning of life at a temple in Tibet. 3. Partial credit is not a bad thing. For ASST2 my partner and I had nothing truly working, but a great deal of code which we believe was not far from what it needed to be. His schedule between other classes and work kept him from being able to make any office hours. I have already covered why I would not bother with office hours. We gained no points for ASST2, despite putting in honest work. Who knows, maybe we were just one small

piece of syntax off from scoring well. 4. Was a Core Map ever actually discussed in lecture? Ever?

- ~~Frangiles~~ test161 suite available sooner and clarity on a few requirements for the homework, starting conditions, sbrk, etc.
- I think the only thing that may need improvement is that the assignment requirements should be laid out in more detail explaining what files need to be changed and the scope of each assignment. Currently the descriptions especially with asst 3 seem to general and brief.
- Should make an course website quickly before starting the semester. Hard to study and catch up in the first two weeks even the materials for the course are hard.
- assignments are good but its not possible to do them in a time line provided specially if you also have other subjects that require project submissions.
- Not much to suggest other than get the course format down. This year's format was a new one, so at times things were a bit questionable.
- Only thing I could think of would be to spread out assignment 1 and 2 together, just so that there is not such a huge jump from the first assignment from the next. Not sure if this is possible given the material but if it could be done I think it would help prepare the students more.
- It's truly difficult to provide constructive feedback for this course. In writing, the class is spotless; We as a student body could not ask for a lecturer with a better wealth of knowledge and determination than Geoff, or a course staff so willing to give countless hours per week in aid on assignments, or a more awesome website that ties the class components together. That being said, it is clear that this class was catered for a school in which the courses leading up to it prepared students in a way that implementing concepts covered in the course is not an entirely new task for them. The course is awesome, and it's just unfortunate that it did not fit into UB-CSE's current curriculum that well. The specific elements of the course that may require some work in my opinion, are:
 - Assignment descriptions: Each assignment released to students seemed to get more vague than the last in terms of their explanations. By the last assignment (ASST3), it was difficult to understand all the moving parts and decipher what was being asked of us, propounding the level of confusion and idea of where to start the assignment.
 - Lecture Depth: As mentioned above, the lectures in the class were brilliant and engaging. What became frustrating though, was the level in which they related to assignment progress (or lack thereof). Even after attending every lecture, and revisiting the notes/slides online, it seemed difficult to apply things learned in class to developments on the projects. I understand that code should not be covered in class (or recitation for that matter) and that there is only so much time to relay material, but there was a clear disconnect between the theory/concepts learned in lecture and what was required to implement in the projects that made it difficult to develop momentum in assignment work.
- I would suggest adding back in the grades for the code reading questions in the future as I know of many students who neglected to do the code reading due to lack of motivation and I personally found the code reading to be beneficial.
- Before an assignment is distributed, gear one lecture's material towards helping the students fully understand what is expected of them / where to start.
- None
- Setting intermediate deadlines (like for project 3) and giving a little more structured project descriptions could be helpful. Maybe a longer time for project 2 and less for project 3 would have been helpful, IMO.
- The assignments were VERY difficult and time consuming, both of which were completely expected going into the course. However, on the previous page I indicated that the course could have been more organized, and I believe this lack of organization added to the difficulty in completing the assignments. Firstly, the various outlets for information regarding the course should be less "scattered". By that, I mean that we relied on Discourse too much for providing IMPORTANT information that would seem to be better posted on the course web page. My next point provides a little more detail... Secondly, the assignment descriptions on the course webpage do not seem to be organized well enough (or with enough detail) for many of us to actually figure out what to do on the assignments! I know that part of each assignment is figuring out what to do (not JUST doing it), but it sometimes took WEEKS of consulting Discourse, the Assignment description, Jinghao's blog (I know this isn't official course material), and office hours just to figure out what to do, much less actually do it. To me, this means that the assignment descriptions are not clear enough, and having to consult so many sources, some of which conflict with each other (more on this in my next point) led to disorganization overall. It seems that the os161 website is intended to be used by anyone who wishes to learn about Operating Systems, regardless of whether or not they are enrolled in a class at UB. However, I do not see how anybody "on the outside" could be successful in doing these assignments without some of the information on Discourse. Things that pop into mind off the top of my head are specifics of how to configure os161 and test161 at the beginning of the course, as well as better descriptions of the workflow for each assignment. It would have been helpful to know more details of dependencies in the tests and circular dependencies in the functionality of the operating system components. A little more detail about a relative order of what to implement would have gone a long way, and I feel that this is something that should be found on the assignment descriptions, rather than searching through Discourse (if it was even on Discourse). It took a LONG time and a lot of crawling around in the dark until we could get to a point where we say "Okay, we need to implement functions A, B, and C, and then tests D, E, and F should work. Then implement data structure U with functions V and W, and tests Y and Z should work." Some more detail to get us to that point would have been very helpful. I made a point in my previous paragraph about sources of help not being consistent. I definitely found this to be true. I know that when asking TA's/ninjas for help, we are asking different people and we will of course get different opinions. However, my partner and I experienced instances where we went to office hours and were told information that was flat out WRONG. I know that TA's/ninjas took a different version of this class than we are currently taking, but every wrong piece of information that's given is a huge setback in a course that I'm sure most people would agree is already incredibly difficult. So, the extra office hours that compensate for the difficulty of the assignments are not exactly always fair compensation. A point about all of the office hours that are offered for this course: I appreciate the abundant office hours and especially the willingness of the ninjas to dedicate their time for no formal compensation or credit, but what does it say about the course itself that so many office hours are needed? I don't think I'm wrong in saying that there are about 5-6 times more office hours offered for this course than there are for any other course at UB. So many office hours should not be needed! Since they are, this brings me back to the point of course organization and the lack of detail in the assignment descriptions. If there was more detail and organization in the assignment descriptions, then we would be less

reliant on (possibly conflicting) TA's/ninjas in office hours. Lastly, depending on which recitation one was in, they may not have started receiving any recitation help on an assignment until a week before it was due. (And not the 2nd recitation for the assignment on the actual due date.) This was addressed by posting the notes online after the earliest recitation was completed. I was happy with this solution and would suggest that in the future, the posting of notes keeps being done, or recitations get "bumped back" a week so that they fall more in line with when people need help on the assignments. Thank you for taking the time to read my lengthy response!

- The assignments are difficult. However, this would not be a complaint of mine if we were given more guidance at the beginning of the assignment. I've had to go to office hours over 20 times this semester. This was mostly for questions like "what am I supposed to do next?" or "where do I start with this assignment?". Another comment about this course is the requirement of "C". C was not taught in previous classes, but it is required for the assignments. I have spent way too much time working on the assignments and getting nowhere. I tend to spend a lot of time on the assignments when I fully understand the concept of the assignment but the structure of the assignments are misleading and confusion. So, I feel like the work put into the assignments tend to be detrimental towards the purpose of the task.
 - To not expect students to be great programmers, and assume that they received enough background in programming to complete the assignments. Class requires a lot of C programming, whereas no class has reasonably required nor taught sufficient amount of C to prepare students prior to OS.
 - Having even more office hours. And not necessary all between the hours of 9-5. Or instead, have mentors assigned to each group and larger groups.
 - I find several things unfair about this course: one is the incredible workload. Yes, I've learned a lot about C programming (when will I use that again?), synchronization primitives, processes, address spaces, etc. But I've gone through hell to get what knowledge I've acquired so far. The programming assignments are extremely difficult and seem great for maybe a second course in OS, not an introduction. It's great for people who already like systems programming and have a knack for it, but awful for the rest of us. Also regarding assignments, I think it's unfair to be releasing tests midway through the submission period, especially the day before the due date, as in assignment 2. That's pretty unprofessional. As for assignment 3, several groups who managed to finish earlier than others could submit and get full credit while those who get finished later are subject to newer tests. And too much of assignment 2 was riding on getting `execv` to work. Why did almost every test need to be run from the shell? Even if the file system calls worked perfectly, you couldn't get a decent amount of credit for them until you finished all the process system calls. Have some mercy. Ok, that's a lot of complaining about the assignments. And I know you won't simply replace them with something else. But please, think of ways to not make them a nightmare for most students. Maybe reinstate some code reading questions for credit. I think a lot (probably even most) of the students in the class are discouraged by how things have gone so far. Lecture attendance has plummeted. I've seen less than 20 students in a lecture... You'll have to achieve a better balance if you want most of your students to leave with a good impression.
 - For the coding assignments, simple and clear instructions would go a long way to eliminate confusion. Half the battle of the assignments is trying to understand what we need to do from the website and I'm not talking about the code reading. The directions would lead me and my partner to think we'd have to code one thing, but the tests would test us on other things. It lead to a lot of wasted time. Once we did figure things out, coding wasn't too bad. Also, if there are bits of information vital to understanding the assignment being posted on discourse, it would be awesome if it was grouped with the information on the course website. Along those same lines, another factor that made things confusing is that coding vocab wasn't consistent or clear enough. It's okay if you use synonyms, you just got to be clear that they mean the same thing. It's hard to guess what coding vocab means and google can only help so much. It would be cool if the lectures related to the code a bit more. I didn't find them super helpful for the coding assignments. Also, I felt the organization of the content was a little weird or not as clear as it could have been. Again, vocab was used that wasn't super clear, or vocab was given multiple definitions. If you're trying to build on previous concepts make sure you lump the entire definition or concept together at one point to make sure it's entirely clear to us what you mean. It would also be awesome if you put more coding snippets in the lectures once in a while. It's not always easy to see how these higher level concepts translate into code. I didn't find the recitations to be useful. A couple of the videos from previous years were great, but this years videos weren't super helpful.
 - More details on how to approach assignments and where to start them could be beneficial in class. For instance, I would have benefited from some C/Linux examples in class during the system call portion of the class.
 - Perhaps tweaking the deadlines for the assignments... Make asst. 1 be a one week assignment, give two weeks for file sys-calls, two weeks for proc syscalls, one week for coremap, and three weeks for the remaining two assignments.
 - Do not penalize for memory leaks or penalize less
 - Change the assignments. They take far too much time from the students schedule and most of the class doesn't finish them anyway. You just cannot expect students to start writing kernel level code with the preparation they have from their previous years of CS study at UB.
 - This is a good course, but it is not a 4-credit course. I've had 8-credit courses that were half as much work as this class.
 - More general guidance on thinking and approaching ASST3 - very little resources exist. Midterm needs ~10 minutes more of time. Major issue: tests (for test161) need to be FROZEN a minimum of a week before assignment due date - sans any bugfixes. It shouldn't be possible for me to pull changes two days before the due date and fail a test, due to an additional check being added.
 - The coding assignments need be explained more or just made easier
-

- Other (please specify) Heard we would learn a lot about operating systems, word of mouth
 - Other (please specify) Interest
 - Other (please specify) I was interested in learning concepts of OS.
 - Other (please specify) Wanted to see how it would be to take one of the best courses at UB.
 - Other (please specify) I wanted to know basics of Systems
 - Other (please specify) Want to learn something about OS
 - Other (please specify) Was interested in how operating systems work/ structured.
 - Other (please specify) Out of interest and the profs reputation
 - Other (please specify) I was looking for really good hands on OS course
 - Other (please specify) Interest
 - Other (please specify) Its a challenging course
 - Other (please specify) Interested in the subject
 - Other (please specify) to learn more about OS
 - Other (please specify) Needed a 300+ CSE and heard professor was great
 - Other (please specify) Interest in mastering OS concepts
 - Other (please specify) Interested in the course.
 - Other (please specify) Interest
 - Other (please specify) The course was interesting and challenging
 - Other (please specify) Assignments
 - Other (please specify) Was interested in the subject.
 - Other (please specify) I like systems
 - Other (please specify) It's a suicide mission.....
 - Other (please specify) Operating Systems was why I developed an interest in the field
 - Other (please specify) good word of mouth
 - Other (please specify) Required, but I also heard that the spring version was the superior course, so I took this one specifically.
-

- nope
 - No Labs available so I had to buy a laptop if I wished to get help from TAs and use office hours
 - n/a
 - The teacher recommends that we work on our assignments in office hours. But the office (Davis 303) can only comfortably sit 6. At times, there are up to 4 Teaching Assistants and I've seen 15 students at office hours. This clearly did not work very well.
 - N/A
 - Office hours many times are overcrowded, since the assignments are impossible to complete without help from TAs and ninjas. There needs to be more space for them. Maybe reserve a classroom, at least for some of the time?
 - course staff is available for office hours but this can be managed better. as there are long queues to meet TAs specially during assignment deadlines
 - Difficult to work and collaborate with TA staff without a laptop.
 - More space needed in TA room
 - Towards the beginning of the semester it was easy to go to TA office hours and get help, but towards the end you were lucky to get a seat or ask a few questions. I was thiking maybe 338 ([A] I think, the bigger room) would be a cool spot. That way the TAs could write on the white board and it be up for when another team needs help. Its also a nice collaborative circle with a projector.
 - Bigger room for TAs. Because there would be lot of students during TA hours.
 - The TA space and a lack of a proper system when meeting with the TA's during office hours
 - Need more space in the TA space
 - I was satisfied. Some more chairs at the TA area would be nice(r).
 - NA.
 - Some of TAs were not much useful as they didn't provide help other than saying debug it. (We come to you after debugging for hours and to have another look at our code / design to see what did we miss ! we know debugging is the key to problem - we don't come to TA hours to have such answer.) But other TAs and ninjas were really good and cleared our doubts within minutes and unblocked us. That being subjective, I would say more space and more TA time should be made available.
 - TA's given small room in which to help students individually. Bigger space needed for office hours
 - The lab space had no room for more than a coupe of people. For the amount of work required and the office hours there should be a bigger room for more people to work.
 - Geoff had to give up his personal office in order to provide space for TAs. It was still all too cramped a lot of the time. Davis Hall may already be too small.
 - We need more space for office hours, not just this class but also in general. The third floor of Davis is always a circus and all courses would benefit from more space.
 - A bigger separate room would be nice.
 - waited till the deadline was due to give out any clues Recitation
-

- He taught well, and made class exciting.
- He is awesome, great personality, knowledgeable, and enthusiastic about the material. I believe some aspects of his class are too difficult though. Maybe I just did not prepare myself for the workload this semester, but I was under the impression a 4 credit class would reasonably take 8-12 hours of preparation outside of class per week. The required time to complete the assignments easily doubled or tripled that each week. I understand that in Geoff's mind, an insane amount of time is not a requirement to completing the assignments, but when the only undergraduate groups I know that got full points before the deadlines admitted to spending hundreds of hours on them, it may as well be a requirement. In a situation like that, it is easy to feel like your performance is not being judged by your ability to grasp the concepts expected by the course syllabus, but by your willingness and ability to spend inordinate amounts of time on a single course. However, I do understand that Geoff wants the course to be challenging and worth the effort. I can say that every hour put into this class is worth it in terms of personal growth and learning, just not everyone can afford to put in a lot of hours. Geoff was flexible in allowing late submissions, and the curve (based on the midterm grades) was generous. Overall though I feel as though students would be more receptive to learning when more manageable goals are laid out before them.
- Dr. Challen (or gwa as he prefers) presented the material of the course in a way that was engaging, memorable, and funny (far more than a topic as dreadful as OS structural implementation has any right to be), which greatly helped on examinations.
- His enthusiasm for the topic was greatly reflected in his method of instruction; he explained topics quite thoroughly and made learning the material quite enjoyable
- Enthusiastic and clear professor. Clearly spent time putting together course materials and designing infrastructure of course so that technical issues would not hold us back from working on assignments. His personal assistance was accurate without giving away answers.
- Very
- Ver Effective
- Very stimulating, helped instill passion in Operating Systems
- He was a very good lecturer and kept the course interesting. The only bad note is I feel that more information could have been given on the projects either in lecture or recitation.
- Great professor. He's a little intimidating but that just made me use every other possible resource before going for help
- The instructor was very effective in explaining the course material, but not very effective in making a connection to the practical, how the lecture pertain to the programming aspect of the course.
- Dr. Challen was interesting, knowledgeable, and challenging.
- I felt like only smart questions should be asked, and I felt very insecure in asking dumb questions, this is a general feeling in the computer science department and I believe should be addressed. Some students who felt like they were behind, only got more behind, because of fear of scrutiny of the TA's, other students, and the Professor.
- He taught the course really well. I liked his approach basing the entire lecture as a series of questions
- He is really enthusiastic about this course.
- He made students feel very unwelcome during office hours, he seemed hostile and rude at times to students who were struggling but nice to others who had a better understanding. I was personally scared of interacting with him due to the way he treated students during office hours.
- He is the best computer science teacher I have come across in my life so far!
- Very effective, the instructor and his TAs did a lot of work with the assignments and it showed in how meticulously test cases were pushed and graded
- He was awesome
- He is awesome ! I loved his teaching style, I was already interested in systems and OS, have worked in device drivers before. But he provided another angle to how to design systems and tradeoffs. He is one of the best teachers I have listened to the date.
- Clear mind and interesting talk
- GWA was good. He loves teaching this course. I learnt quite a lot by taking CSE 521.
- He aced the course PERIOD
- Proff GWA is really good and knows his stuff. His notes and lecture slides and videos really helped me learn the subject. :)
- Carefully explains material during lectures and answers student's questions, active on discourse too.
- GWA is AWESOME! He puts in a lot of work into the course and expects the same from his students. He makes it awesome environment to learn in.
- Excellent
- Definitely one of the best instructors at UB CSE, it seems to become harder and harder as every year it seems that the instructors are required to teach more students. Just having a professor that does stand in one spot reading off words from a 20 year old powerpoint is sufficient for me. GWA was constantly moving, asking questions, reviewed material every day and most importantly knew the subject, talked about it before he even got to the slides which made it a lot easier to pay attention.

- Good.
- A former professor. Although I didn't attend many lectures, he posted videos which were priceless. Geoff really W a lot from his lectures and assignments.
- Geoff was extremely effective in this course. The arrangements for help in assignments was incredible and really helped us in the projects
- The enthusiasm that Challen exhibited was amazing. I really felt bad when I missed every single class. I never felt bored in class. The assignments were well supported by the course staff, the opportunity to submit multiple times allowed us to experiment a lot. The 3rd assignment could've given less time for coremap and more time for other two parts. Also, the targets could've been released early so that people could've got more time to test.
- Challen was very effective and knowledgeable in his explanation of concepts. His enthusiasm showed in every class. He truly likes to teach Operating Systems.
- Awesome
- instructor assumes everyone has basic OS knowledge when teaching while that is not always true.
- The prof was really good. Had loads of fun.
- The instructor presented the material in an organized, easy-to-follow, and entertaining manner.
- I have no problems with this course other than having extreme difficulty with programming in c which I cannot fault the course staff for.
- Would have been better if the project work was stepped through as a class and the grading was shifted to frequent assessment of the understanding of the material resolved in the project.
- Very effective
- GWA was good, but he can be intimidating for newbies. I wanted to seek help, but was not particularly keen due to his attitude. I went to the TAs instead.
- Geoff made me want to come to lecture because of his engaging teaching style. He is entertaining and crams tons of information into each lecture, but it is very easy to digest.
- He is very enthusiastic in teaching the concepts which keeps the course engaging.
- Very effective. I liked how he used anecdotes throughout the lectures.
- I found no faults in this course other than it being difficult, but in a good way.
- There are 2 sides to Geoff either he is in a great mood or he is in a bad mood. I think that he is generally a really nice guy but you do not what to get on his bad side.
- Good, likes to say "right" a lot but he got better at it. Enthusiastic, likes to make jokes and funny during lecture.
- He is based on the slides that he made. And it is very effective! :)
- He was very effective. Maybe the only thing would be to give more of a background on the assignments and more direction in doing them. Given the size of the assignments, it seemed very difficult to have to rely on the TAs for instruction on how to do them even with a plethora of office hours.
- Geoff is an excellent lecturer, with a clear passion for the field and determination to help students earn knowledge. It was very evident from the beginning that Geoff 'knew his shit' and that his primary goal was to instill wisdom. On top of that, he seems great at managing his course staff and continuously updating and buffing out his course for the better. Geoff clearly doesn't like to be bugged by monotony; Like any professor at a monstrosly sized school, he doesn't want to address the same stupid questions over and over again. He does, however, handle being outnumbered very well and implements many systems (course staff hours, discourse, etc.) to get everyone the assistance they need. That said, he still made the effort to help and encourage individuals in need. Sure, he's abrasive, but neither a good bedside manner nor hand-holding will prepare one for a challenge or life itself. All in all, it would be a pleasure to work with him or take another course of his again.
- Geoff was excellent in this course. Giving me the correct balance of well presented material, well defined goals, and solid guidance when seeking help in office hours.
- The lectures were among my favorite during my time at UB. But, the assignments were among my worst and most time consuming assignments.
- I enjoyed having Geoff as a professor for this course! To me, Geoff was very effective in teaching the conceptual aspects of the course, and I enjoyed attending the lectures. There was a disconnect in the difficulty of the conceptual aspects of the course and the programming aspects. This disconnect could have been alleviated by a more directed set of assignment descriptions, as stated earlier in my evaluation. This is my only "complaint", and it stems from the amount of struggling I did on the assignments. I have heard that CSE421/521 is a little above the level of many UB students REGARDLESS of who teaches it. I think there is a systemic problem here: either the 421/521 course itself, or the way UB's curriculum up to this point has (not) prepared us for this class. Everything being said, I would completely recommend Geoff for any class. His enthusiasm, willingness to help, and the way that he makes the entire course (notes, lectures, recitations, etc.) available online show that he is dedicated to students!
- Good at teaching material, not that good at writing what should be done for projects.
- Geoffery was very effective in teaching the class. He presented the ideas in a way that was easy to understand while drawing interest with current applications of the material.
- Engaging lecturer
- Great lecturer. Explains concepts very well. The only drawback about the course is the unrealistic expectations of having UB students complete an assignment that is same as Harvard. We are not, and never will be Harvard students. We very likely did not receive same preparation as Harvard students have, nor have the same life circumstances as they very likely have.

- Geoffrey was very effective at delivering material for exams but prepared us 0% for any of the assignments leaving the lackluster result to TAs in recitation.
 - Probably the best professor I'd had since my first CS teacher who gave me the passion for the field to begin with. The conversational style of the lectures works amazingly well to get across what are often somewhat abstract concepts.
 - Very enthusiastic and interested in the material
 - Geoff is good at giving lectures (sometimes it's a little fast...) But I don't understand why it's necessary to berate students when approached for help. I've seen and heard this happen on multiple occasions (and from secondhand accounts). There are much more constructive ways to motivate students and help them learn the material than by putting them down for their mistakes.
 - It is so hard for me...but the thing is installing os was not easy
 - Very passionate about teaching - I can tell he cares a lot about the course
 - This is by far the best professor I've had at UB and really makes the course an enjoyable experience. He's super knowledgeable when it comes to course material, and you can tell he enjoys teaching it. The course is definitely hard, but his office hours are inviting and helpful, and he's always positive and funny. He's dedicated to the learning of his students as well as the entire CSE department.
 - Lectures were pretty dope. They were enough to understand the higher level concepts and for written stuff, but not good enough for the coding assignments.
 - Geoffrey Challen beyond surpasses the cumulative quality of the CSE faculty. His integration of humor within course material was profound; indicative of a humble, superior level of intelligence. I can't articulate how appreciative I am for the accessibility ease of lecture material. Some people are not fortunate enough to live the "full-time residential student" life, and it's extremely discouraging when attending to responsibilities results in falling behind/missing announcements in classes. TLDR: 10/10 would recommend would tell a friend.
 - Instructor Challen was not effective at teaching this course. He recommended that we did not need to get the book for the course, but then expected us to have in-depth knowledge of the course to do assignments. Instructor Challen was very condescending toward students inside and outside of class. Not very approachable. Relentlessly talked about his time at Harvard as a way to belittle students and reduce their morale. Instructor Challen was not effective at teaching this course.
 - He was very good in his teachings. Also, he IS cute :)
 - Unapproachable. Very knowledgeable though. But expects you to already know too many things.
-

7. Can you comment on how effective the teaching assistants(s) were in helping you meet the learning outcomes of the course. -

- great
- Ninja namedrop: Mark and Issac were the most helpful. Some I did not meet because our schedules conflicted, and others I did not bother because they used the TA hours to bring in their own homework and thwart questions to official TAs. Regarding the TA that was dismissed: Yes, code sharing was explicitly prohibited, but I did appreciate the notion of providing implementation examples (or rather, examples of what *not* to do) in the beginning of the course, as the structure of OS161 can be intimidating, and confusing to learn at first. Regarding Scott: Wow. That's a good wow. Incredibly helpful and inspiring.
- Very effective..
- Really Effective. They know their stuff
- Extremely, undeniably, remarkably effective. This course couldn't survive without them and their devotion to the students.
- They were helpful in office hours, but recitations weren't super helpful.
- Office hours were extremely helpful for working on assignments
- Good power points
- The course TAs/Ninjas were an invaluable asset throughout the entire semester. Any conceptual and/or debugging question that I asked I got enough information and insight into to solve the problems I was having after being stuck.
- First off, this course has a plethora of TA's that are active both in office hours and on the course forums. They are very good at not directly giving away answers, but helping in a way that pushes you in the right direction. While office hours could get packed at times, the TA's were very good at handling these situations and getting students to look at their problems in a way they may not have thought of before.
- Certain TA's (and course staff) were extremely helpful - often going the extra mile. However, the problem is the inconsistency between course staff. For example, eventually I would stop going to everyone's office hours except for specific course staff - because of how unhelpful I (and other friends in the course) found certain staff members
- they do their best job
- They were great. Undergrad TAs are exceptional students, and senior TAs are very helpful.
- TAs are good.
- The official teaching assistants generally provided very good advice for the assignments. The ninjas were hit-or-miss, unfortunately.
- The TA's were very friendly and helped in any way they could.
- There were always opportunities to meet with TA and get help on assignments. They helped figure out how to lead you to an answer and not just fix things. They almost always answered my questions to the point where I understood.
- They are not Tibetan monks. Let them off the leashes and actually help. Some students actually only come to ask for help when... they actually need help.
- Ali was excellent, that first TA was no good.
- The teaching assistants were extremely helpful for this class. It was nice to have such a helpful group of individuals around to help for such a difficult class.
- Many of the TAs knew the material well and were very helpful when asked.
- The teaching assistants did an outstanding of helping me and my partner understand the material.
- The teaching assistants were helpful when I seeked help with the assignments.
- Same as Geoff. Based on slides when they teach and helped us alot in office hours
- Classic group of TA's. Just trying to balance what their boss tells them and what their student body wants. Did a great job at it.
- They are required to complete this course. I don't see how anyone could complete this course, given that it isn't their only class for the semester, without the help of the TAs. I felt completely lost without them giving me the direction on where to start and where to go. Overall they were completely depended upon and did a fantastic job exceeding TA expectations.
- EPIC, while I may fail this class I wouldn't have even tried if not for the aid of the course TAs.
- I think there is a lot of variance in the effectiveness of the TAs. Some of the TAs were great and would spend hours helping students. But there were others who would try to find issues in other places instead of answering the question at hand.
- Teaching assistants were approachable and available
- Extremely. Some were more knowledgeable than others. If a ta is the only Ta in office hours, He must be trained better
- They were really helpful.
- TA's are often able to get students "unstuck" during assignments.

- Very helpful
- Extremely important. they are the reason people pass the course.
- We need more space in TA room and some more TA
- Always available. AWESOME work!
- Since this was the first time that test161 was introduced, there were few race conditions that resulted in some minor inconsistencies, but it was not a major issue, but needs to be rectified.
- Challen has a team of TA's and Ninjas without whom it would have been very hard to finish the assignments at all.
- The recitations conducted were very helpful. Also the TA support is the best I've seen in UB so far
- Special thanks to Carl, Ben, Scott and Mark! Each spend a great deal of time with me answer questions and helping debug code. They also stayed well past their time slots on some days.
- Tis were very helpful.
- Very much.
- Apart from one rude ninja, other ninja's and TA's I have contacted were very helpful. I got a lot of concepts cleared and got useful hints too. The rude ninja did not want to listen to what I had to say and kept shouting on me while what I was trying to ask was entirely different and what he understood was completely different.
- TAs WERE GREAT! they helped incredibly well on the assignment, and any other questions I had.
- Some are helpful some are not
- Some TAs obviously were better than others. The biggest issue was the sheer mass of students at office hours and the overall lack of one on one unless you are prepared to sit at office hours for hours every day. The assignments need to be made easier to lessen the gap in knowledge in being able to do the assignments lessening the load on the TAs. This course had an insane amount of office hours available and no matter when I went it was packed leaving any chance for help low.
- Very effective
- Pretty useless as they were not allowed to give us many hints
- Can deal most of the troubles we meet in assignment. Give useful thoughts/ideas instead of telling the solution directly.
- They were very good. OS161 is complicated and very few can master it but they did a good job.
- Many of them, mostly effective
- Mentioned already in previous stage. Some were really good, some were bad and have attitude. If you can't explain what's wrong with the design or understanding of student - what's the point of being a TA ? If student knows everything as some TAs expect, why they would come to you !
- There are many TAs(apart from official, some extra TAs are also there) so it becomes very easy to seek help because the office hours are very frequent. Also, the TAs are very knowledgeable so they really help in completing things and solving difficulties.
- There were many TAs and many office hours which is useful and necessary for this course. Most TAs would spend as much time as was needed to fully address my problems. Some TAs relied on their solution code heavily rather than actually discussing the problem at hand. There is little reason to expect that a mismatch in our code is the way to fix our problem. Also TAs would often give contradictory advice. This is expected to some extent due to the nature of the course but sometimes it was clear that misinformation was provided.
- very helpful, should be a class led by the TA's
- The TAs are pretty much required to both understand the material and implement what is necessary for the assignments but they are readily available to do so.
- If the TAs hadn't been there to help, this course would have been impossible to pass!
- The teaching assistants were helpful in answering general and assignment specific questions.
- They were very helpful for the assignments. But time and space management for office hours has much scope for improvement!
- waited until the dead line to give a clear explanation of the project.
- Easily the best TA's I've ever interacted with.
- They avoided answering questions asked.
- The TA's were absolutely invaluable when it comes to the course; they explained things quite thoroughly and, for the most part, helped quite a good deal
- The TAs were always either able to provide immediate support, or point you in a direction to do so.